



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Технологический институт сервиса (филиал) ДГТУ в г.Ставрополе  
(ТИС (филиал) ДГТУ в г.Ставрополе)**



**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
по выполнению курсовых работ по дисциплинам  
для студентов направления подготовки  
09.04.02 Информационные системы и технологии  
программа магистратуры «Информационные системы и технологии»



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Технологический институт сервиса (филиал) ДГТУ в г.Ставрополе  
(ТИС (филиал) ДГТУ в г.Ставрополе)**

УТВЕРЖДАЮ

Директор

\_\_\_\_\_ Е.А. Дрофа

\_\_\_\_\_ 2022 г.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
по выполнению курсовых работ по дисциплинам  
для студентов направления подготовки  
09.04.02 Информационные системы и технологии  
программа магистратуры «Информационные системы и технологии»



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Технологический институт сервиса (филиал) ДГТУ в г.Ставрополе  
(ТИС (филиал) ДГТУ в г.Ставрополе)**

## **МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

по выполнению курсового проекта  
по дисциплине «Анализ и синтез ИС»  
для студентов направления подготовки  
09.04.02 Информационные системы и технологии  
Направленность (профиль) Информационные системы и  
технологии

## Содержание

Введение

Цель, задачи и реализуемые компетенции

Формулировка задания

Структура работы

Общие требования к написанию и оформлению работы

Последовательность выполнения задания

Критерии оценивания работы

Порядок защиты работы

Список рекомендуемых информационных источников

## **Введение**

В настоящих методических указаниях представлена методика написания курсовой работы по дисциплине «Анализ и синтез ИС» для студентов, направления 09.04.02 Информационные системы и технологии. Методические указания по проведению к курсовой работе разработаны в соответствии с рабочей программой учебной дисциплины.

Курсовая работа позволяет развить творческий потенциал студентов и подготовить их к выполнению выпускной квалификационной работы. Главное назначение курсовой работы по дисциплине «Анализ и синтез ИС» состоит в подготовке студентов к самостоятельному выполнению исследовательской работы, связанной с проблемами в области экономики.

### **Цель, задачи и реализуемые компетенции**

Написание курсовой работы - составная часть учебного процесса, предполагающая самостоятельную работу студента. Цели и задачи курсовой работы разнообразны: научная, познавательная, учебная, методическая. Данные цели проявляются через следующие конкретные задачи курсовой работы:

- систематизация, закрепление и расширение теоретических знаний, полученных на лекциях и семинарах по экономической безопасности;
- привитие навыков самостоятельной работы по подбору литературы, работы с научной литературой;
- умение самостоятельно систематизировать и изложить знания, полученные в процессе самостоятельного изучения литературы;
- привитие навыков научно-исследовательской работы, использование анализа и самостоятельных выводов по экономическим проблемам;
- углубленное изучение наиболее актуальных экономических проблем, уяснение связи теории с практикой в обеспечении экономической безопасности.

В результате освоения данной дисциплины формируется следующая компетенция у обучающегося:

ОПК-5.2: Использует современное программное и аппаратное обеспечение информационных и автоматизированных систем

УК-1.3: Использует методологию системного и критического анализа проблемных ситуаций, методики постановки цели, определения способов ее достижения

Изучив данный курс, студент должен:

Знать:

принципы системного представления основных этапов проектирования информационных систем, основанного на объектном подходе.

Уметь:

использовать промышленные стандартизированные решения, опирающиеся на современные технологии;

проектировать информационные системы от этапа постановки задачи до программной реализации.

Владеть:

методами анализа информационных ресурсов; разработки различных моделей данных; конструирования программных модулей; анализа проектных решений.

Реализация компетентного подхода предусматривает широкое использование в учебном процессе активных и интерактивных форм проведения занятий (разбор конкретных ситуаций, собеседование) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков специалистов.

Курсовой проект должен свидетельствовать о способности автора к систематизации и расширению полученных теоретических знаний, о владении практическими навыками по дисциплине «Анализ и синтез ИС», о сформированности компетенций при решении поставленных в работе задач.

Курсой проект должен привить студенту навыки творческого изучения и решения профессиональных задач.

Проведенное исследование может касаться чисто теоретической проблемы или ориентироваться на решение практических задач, связанных с дальнейшей профессиональной деятельностью студента.

# 1 Основные теоретические сведения

## 1.1 Методы и средства проектирования информационных систем и технологий

**Информационная система** — система, предназначенная для сбора, хранения, обработки, поиска, распространения, передачи и предоставления информации (ГОСТ 7.0—99 п. 3.1.30).

Информационные системы и технологии (ИСТ), как и любые другие системы имеют определенный жизненный цикл - непрерывный процесс, который начинается с момента принятия решения о необходимости создания ИСТ и заканчивается в момент ее полного изъятия из эксплуатации. Жизненный цикл является моделью создания и использования ИСТ [1], которая отражает различные состояния системы с момента возникновения в данном комплексе средств до момента его полного выхода из употребления.

Основным нормативным документом, регламентирующим состав процессов ЖЦ систем, является международный стандарт - ISO/IEC 15288:2008 Standard for Systems Engineering — System Life Cycle Processes (Системная инженерия - процессы жизненного цикла систем). Российским аналогом является ГОСТ Р ИСО/МЭК 15288-2008 «Информационная технология. Процессы жизненного цикла систем» [1].

Среди множества концепций проектирования информационных систем в настоящее время следует выделить разработку, управляемую моделями (Model Driven Development, MDD). MDD - это развивающаяся парадигма, решающая многочисленные проблемы композиции и интеграции крупномасштабных систем и опирающаяся при этом на имеющиеся достижения в области технологий разработки программного обеспечения (в частности, на компонентное промежуточное программное обеспечение). MDD позволяет перевести разработку программного обеспечения на более

высокий уровень абстракции по сравнению с тем, который возможен при использовании алгоритмических языков. Для представления элементов системы и их связей в подходе MDD используются модели. Модели служат входными и выходными данными на всех стадиях разработки, вплоть до генерации законченной системы.

Популярным вариантом MDD является модельно-управляемая архитектура (Model-Driven Architecture, MDA), предложенная и развиваемая консорциумом Object Management Group (OMG). В подходе MDA системы представляются с использованием языка моделирования общего назначения Unified Modeling Language (UML) и его конкретных профилей.

Начинается разработка с создания независимой от платформы модели (PIM). Затем после выбора языка программирования, исходя из специфики разработки, осуществляется трансформация PIM -модели в одну или несколько моделей, определяемых платформой, в рамках которой они реализуются (PSM).

Для разработки информационных систем и технологий по данной концепции могут использоваться различные методы и подходы. Подход к созданию ИСТ определяется набором составляющих его этапов, их последовательностью и используемыми на каждом этапе моделями. Кроме последовательности этапов подходы проектирования отличаются объектами исследования и синтеза. В зависимости от способа декомпозиции системы для проектирования информационных систем используется два основных подхода: *структурный* и *объектно-ориентированный*.

Сущность структурного подхода к разработке ИСТ заключается в ее декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Элементами декомпозиции являются модули, связь между которыми реализуется через



передачу управления. Система представляется совокупностью взаимодействующих модулей или процедур.

В объектно-ориентированном подходе основным элементом декомпозиции является объект, который может быть ассоциирован с объектом реального мира. Объект содержит данные о своих свойствах и состояниях, процедуры для изменения данных и связан с событиями, которые приводят к изменению его свойств. Система представляет собой совокупность взаимодействующих объектов.

Основными средствами проектирования информационных систем и технологий являются:

- языковые средства;
- инструментальные средства.

Среди множества языковых средств в настоящее время широкое распространение получил унифицированный язык моделирования(UML). UML это открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования в основном программных систем. UML не является языком программирования, но в средствах выполнения UML-моделей как интерпретируемого кода возможна кодогенерация. Формальная спецификация последней версии UML 2.0 опубликована в августе 2005 года. Семантика языка была значительно уточнена и расширена для поддержки методологии разработки, управляемой моделями. Последняя версия UML 2.4 опубликована в августе 2011 года. UML 1.4.2 принят в качестве международного стандарта ISO/IEC 19501:2005.

Язык UML имеет специальные базовые элементы графической нотации, которые необходимы для формирования всех модельных представлений. В частности в UML 2.0 существует 13(UML 2.2 - 14)

официальных диаграмм, каждая из которых отражает различные аспекты системы.

Для того чтобы UML 2.0 поддерживал MDD, нужна платформа, имеющая обширные средства для адаптации языка манипулирования моделями. На поддержку MDD претендуют множество коммерческих инструментов UML. Лучшие из них поддерживают, в известных пределах, определение и использование трансформации моделей UML, но они ограничивают пользователей конкретной платформой реализации.

К наиболее популярным инструментальным средствам UML-моделирования относятся Rational Rose (IBM), Together (Borland) и MS Office Visio. Среди указанных средств особо следует выделить Borland Together, встраиваемое сегодня в распространенное RAD средство Borland Developer Studio (BDS), включающее Delphi, C++ Builder и C# Builder. Применение BDS позволяет в одной среде осуществлять как моделирование процессов и систем, так и создания приложений для их реализации на языках программирования высокого уровня.

Таким образом, MDD-разработка, основанная на MDA, в основном занимается трансформацией моделей и генерацией кода. С ее помощью разработчики сначала создают модель объекта на унифицированном языке моделирования, а затем генерируют код из этой UML-модели, применяя инструмент генерации кода. Модели, используемые для анализа и проектирования объектов информационных систем в языке моделирования UML, представляются в виде диаграмм.

## **1.2 Диаграммы унифицированного языка моделирования**

Применение UML 2.0 позволяет разделить проблему моделирования сложной системы на составные части с помощью четырех представлений:

-статическое структурное представление модели описывает структурные аспекты системы, например, с помощью диаграммы классов;

-представление взаимодействия используется для моделирования последовательностей действий и коммуникаций, описывающих кооперацию взаимодействующих экземпляров;

-представление деятельности используется для создания моделей, описывающих поток «деятельностей» в системе;

-представление в виде конечного автомата используется для описания поведения системы в терминах состояний и переходов между ними.

Эти представления не являются полностью ортогональными: концепции, используемые в одном из них, часто зависят от концепций, применяемых в другом. Так, классификаторы участников взаимодействия должны быть определены в статической структурной модели. Такие зависимости определяются в метамодели UML, и инструментальные средства могут их задействовать для определения согласованности информации во всех представлениях системы.

Графические обозначения отдельных элементов моделей будут представлены при создании диаграмм в дальнейшем. Рассмотрим краткую характеристику диаграмм UML 2.0.

Структурные диаграммы

**Диаграмма классов**(Class diagram) — статическая структурная диаграмма, описывающая структуру системы, она демонстрирует классы системы, их атрибуты, методы и зависимости между классами.

Существуют разные точки зрения на построение диаграмм классов в зависимости от целей их применения:

-концептуальная точка зрения — диаграмма классов описывает модель предметной области, в ней присутствуют только классы прикладных объектов;

-точка зрения спецификации — диаграмма классов применяется при проектировании информационных систем;

-точка зрения реализации — диаграмма классов содержит классы, используемые непосредственно в программном коде (при использовании объектно-ориентированных языков программирования).

**Диаграмма компонентов**(Component diagram) — статическая структурная диаграмма, показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами. В качестве физических компонент могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и т. п. Диаграмма компонента показывает структурные отношения между компонентами будущей информационной системы. В UML 2.0 компоненты являются автономными инкапсулированными единицами (unites) внутри системы или подсистемы, которые обеспечивают один или несколько интерфейсов. Поэтому диаграмма компонента позволяет архитектору убедиться в том, что компоненты реализуют заданную функциональность системы.

**Диаграмма композитной/составной структуры** (Composite structure diagram) — статическая структурная диаграмма, демонстрирует внутреннюю структуру классов и, по возможности, взаимодействие элементов (частей) внутренней структуры класса. Подвидом диаграмм композитной структуры являются *диаграммы кооперации* (Collaboration diagram, введены в UML 2.0), которые показывают роли и взаимодействие классов в рамках кооперации. Кооперации удобны при моделировании шаблонов проектирования. Диаграммы композитной структуры могут использоваться совместно с диаграммами классов.

**Диаграмма развёртывания**(Deployment diagram) — служит для моделирования работающих узлов(аппаратных средств) и артефактов, развёрнутых на них. В UML 2 на узлах разворачиваются артефакты (*artifact*), в то время как в UML 1 на узлах разворачивались компоненты. Между артефактом и логическим элементом (компонентом), который он реализует, устанавливается зависимость манифестации.

**Диаграмма объектов** (Object diagram) — демонстрирует полный или частичный снимок моделируемой системы в заданный момент времени. На диаграмме объектов отображаются экземпляры классов (объекты) системы с указанием текущих значений их атрибутов и связей между объектами.

**Диаграмма пакетов** (Package diagram) — структурная диаграмма, основным содержанием которой являются пакеты и отношения между ними. Диаграммы пакетов служат, в первую очередь, для организации элементов в группы по какому-либо признаку с целью упрощения структуры и организации работы с моделью системы.

Диаграммы поведения

**Диаграмма деятельности** (Activity diagram) — диаграмма, на которой показано разложение некоторой *деятельности* на её составные части. Под деятельностью (англ. *activity*) понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов — вложенных видов деятельности и отдельных *действий* (англ. *action*), соединённых между собой потоками, которые идут от выходов одного узла ко входам другого. Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений. Аналогом диаграмм деятельности являются схемы алгоритмов по ГОСТ 19.701-90.

**Диаграмма автомата** (State Machine diagram, *диаграмма конечного автомата, диаграмма состояний*) — диаграмма, на которой представлен *конечный автомат* с простыми состояниями, переходами и композитными состояниями. Конечный автомат — спецификация последовательности состояний, через которые проходит объект или взаимодействие в ответ на события своей жизни, а также ответные действия объекта на эти события. Конечный автомат прикреплен к исходному элементу (классу, кооперации или методу) и служит для определения поведения его экземпляров.

**Диаграмма вариантов использования** (Use case diagram) — представляет собой отражение действующих лиц (актантов), которые взаимодействуют с системой, и реакцию программных объектов на их действия. Актантами могут быть как пользователи, так и внешние агенты, которым необходимо передать или получить от них информацию. Значок варианта использования отражает реакцию системы на внешнее воздействие и показывает, что должно быть сделано для актанта. Основная задача — представлять собой единое средство, дающее возможность заказчику, конечному пользователю и разработчику совместно обсуждать функциональность и поведение системы.

### **Диаграммы взаимодействия**

**Диаграмма последовательности** (Sequence diagram) — диаграмма, на которой изображено упорядоченное во времени взаимодействие объектов. В частности, на ней изображаются участвующие во взаимодействии объекты и последовательность сообщений, которыми они обмениваются. Диаграмма последовательности показывает хронологическую последовательность сообщений между объектами во взаимодействии. Она состоит из нескольких участников, таких как агенты, системы или подсистемы, классы и компоненты, представленные линиями жизни (lifelines), а также сообщения, которыми они обмениваются при взаимодействии.

**Диаграмма коммуникации** (Communication diagram, в UML 1.x — *диаграмма кооперации, collaboration diagram*) — диаграмма, на которой изображаются взаимодействия между частями композитной структуры или ролями кооперации. В отличие от диаграммы последовательности, на диаграмме коммуникации явно указываются отношения между элементами (объектами), а время как отдельное измерение не используется (применяются порядковые номера вызовов). Диаграмма коммуникации показывает поток сообщений между объектами и то, как несколько объектов сотрудничают при

выполнении общей задачи. Как и диаграмма последовательности (sequence diagram), диаграмма коммуникации тоже может моделировать динамическое поведение для варианта использования (use case). Однако диаграмма коммуникации больше нацелена на показ того, как происходит координация, чем на хронометрирование последовательности.

Диаграммы коммуникации и последовательности транзитивны, выражают взаимодействие, но показывают его различными способами и с достаточной степенью точности могут быть преобразованы одна в другую.

**Примечание.**

По причине того, что диаграммы коммуникации и последовательности являются разными взглядами на одни и те же процессы, Rational Rose позволяет создавать из диаграммы коммуникации диаграмму последовательности и наоборот, а также производит автоматическую синхронизацию этих диаграмм.

*Диаграмма обзора взаимодействия* (Interaction overview diagram) — разновидность диаграммы деятельности, включающая фрагменты диаграммы последовательности и конструкции потока управления. Этот тип диаграмм включает в себя диаграммы последовательностей действий и диаграммы сотрудничества. Эти диаграммы позволяют с разных точек зрения рассмотреть взаимодействие объектов в создаваемой системе.

*Диаграмма синхронизации* (Timing diagram) — альтернативное представление диаграммы последовательности, явным образом показывающее изменения состояния на линии жизни с заданной шкалой времени. Эта диаграмма может быть полезна в приложениях реального времени.

Таким образом, выбранный разработчиком набор диаграмм позволяет создать практически любое представление о проектируемой системе.

**Примечание.**

Изображая диаграмму, воспользуйтесь следующими рекомендациями:

- дайте диаграмме имя, соответствующее ее назначению;
- расположите элементы так, чтобы свести к минимуму число пересечений;
- пространственно элементы расположите так, чтобы семантически близкие сущности располагались на диаграмме рядом;
- используйте примечания и цвет, чтобы привлечь внимание читателя к важным особенностям диаграммы.

### **1.3 Унифицированный процесс разработки программного обеспечения**

Процесс проектирования ИСТ, кроме основных концепций и понятий, используемых при проектировании и реализации ИСТ, включает в себя технологию проектирования. Одной из развитых современных технологий является унифицированный процесс( Rational Unified Process,RUP). RUP – одна из лучших технологий разработки программного обеспечения, созданная в компании Rational Software, входящей в состав IBM. Унифицированный процесс позволяет создавать сложные программные системы, основываясь на индустриальных методах разработки [2, 3].

Вся разработка информационной системы (ИС) рассматривается в RUP как процесс создания артефактов. Любой результат работы проекта, будь то исходные тексты, объектные модули, документы, передаваемые пользователю, модели – это подклассы всех артефактов проекта.

Одним из интереснейших классов артефактов проекта являются модели, которые позволяют разработчикам определять, визуализировать, конструировать и документировать артефакты программных систем.

Модели позволяют рассмотреть будущую систему, ее объекты и их взаимодействие еще до вкладывания значительных средств в разработку, позволяют увидеть ее глазами будущих пользователей снаружи и



разработчиков изнутри еще до создания первой строки исходного кода. Большинство моделей представляются диаграммами на унифицированном языке моделирования UML.

Основными моделями, создаваемыми в RUP, являются: модель вариантов использования, модель анализа, модель проектирования и модель реализации. Эти модели являются результатом основных работ процесса, к которым относятся: определение требований, анализ, проектирование и реализация. Рассмотрим содержание каждого из них.

### **1.3.1 Определение требований**

Одним из важнейших этапов разработки ИС, согласно RUP, является этап определения требований, который заключается в сборе всех возможных пожеланий заказчика к работе системы. На данном этапе в ходе интервью с пользователями и изучения документов, аналитики должны собрать как можно больше требований к будущей системе, что не так просто, как кажется на первый взгляд. Позднее эти данные должны будут систематизированы и структурированы. Для того чтобы верно определить требования, разработчики должны понимать **контекст** (часть предметной области) в котором будет работать будущая система.

### **Определение контекста информационной системы**

Для определения контекста ИСТ выполняется предпроектное обследование предметной области(области использования ИСТ). Для этого создаются модель предметной области и бизнес-модель, что является различными подходами к одному и тому же вопросу. Часто создается что-то одно: модель предметной области или бизнес-модель[2].

Отличия этих моделей в том, что модель предметной области описывает важные понятия, с которыми будет работать система и связи их

между собой. Тогда как бизнес-модель описывает бизнес-процессы (существующие или будущие), которые должна автоматизировать(поддерживать) система. Поэтому кроме определения бизнес-объектов, вовлеченных в процесс, эта модель определяет работников, их обязанности и действия, которые они должны выполнять.

Использование UML не ограничивается моделированием программного обеспечения. Его также используют для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

Для создания модели предметной области с помощью UML используется обычная диаграмма классов, однако для создания бизнес-модели ее уже явно недостаточно. В этом случае применяется диаграмма вариантов использования с использованием дополнительных значков, которые отражают сущность бизнес-процессов – это бизнес-актант, бизнес-прецедент, бизнес-сущность и бизнес-управление. Эта модель намного ближе к следующей модели, создаваемой в процессе разработки – модели анализа.

При моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений часто используются диаграммы деятельности.

На практике диаграммы деятельности применяются в основном двумя способами:

- для моделирования процессов;
- для моделирования операций.

В первом случае внимание фокусируется на деятельности с точки зрения действующих лиц, которые работают с системой. Важным здесь является применимость диаграмм деятельности для описания бизнес-процессов. В данном случае для построения диаграмм деятельности используется так называемая траектория объекта, или поток объекта(object flow). Суть его состоит в том, что на диаграмме кроме деятельности можно изобразить и объекты, относящиеся к деятельности. С помощью символа

зависимости(пунктирная стрелка) эти объекты можно соотнести с той деятельностью или переходом, где они создаются, изменяются или уничтожаются. Траектория объекта позволяет показать объекты, относящиеся к деятельности, а также моменты переходов этих объектов из одного состояния в другое.

Рекомендации по построению диаграмм деятельности для моделирования процессов заключаются в следующем.

Моделируют бизнес-процессы в несколько этапов, первым из которых является разбиение их на подпроцессы. Подпроцессы, являющиеся "участками большого процесса", описать легче.

Дальше выделяют ключевые объекты (и создают для них дорожки), определяют предусловия и постусловия каждого процесса (т. е. его границы), описывают деятельности и переходы, отображают на диаграммах состояния ключевых объектов, в которые они переходят в ходе процесса.

В итоге создается не какая-то абстрактная диаграмма, а модель реального бизнес-процесса в реальной компании, занимающейся реальным бизнесом.

Пример детализации конкретного бизнес-процесса с помощью диаграммы деятельности, созданной в Rational Rose, показан на рисунке 1. На диаграмме отражена деятельность выдачи товара со склада [1].

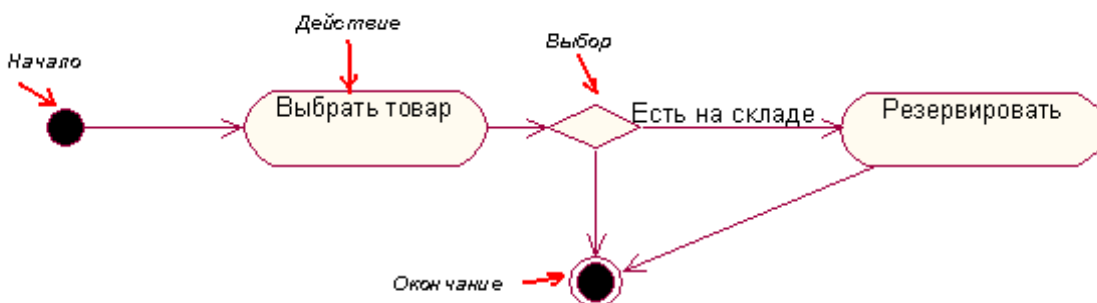


Рисунок 1 - Пример диаграммы активности, созданной в Rational Rose

Второй пример моделирования бизнес процесса оформление заказа в Интернет-магазине представлен на рисунке 2.



Рисунок 2 – Оформление заказа в Интернет-магазине

На рисунке 3 представлена диаграмма деятельности, выполненная в Borland Together. Здесь показана параметризованная деятельность с объектным узлом параметра, соединенным с контактами действий. Объектные узлы параметров размещаются на границе диаграммы, и ребра потока объектов соединяют их с контактами. Тип объекта, удерживаемого в объектном узле, обычно отображается в метке этого узла. В данном примере информация, используемая для заполнения заказа, предоставляется как входной параметр и передается в действие по вызову работы “Заполнение заказа”.

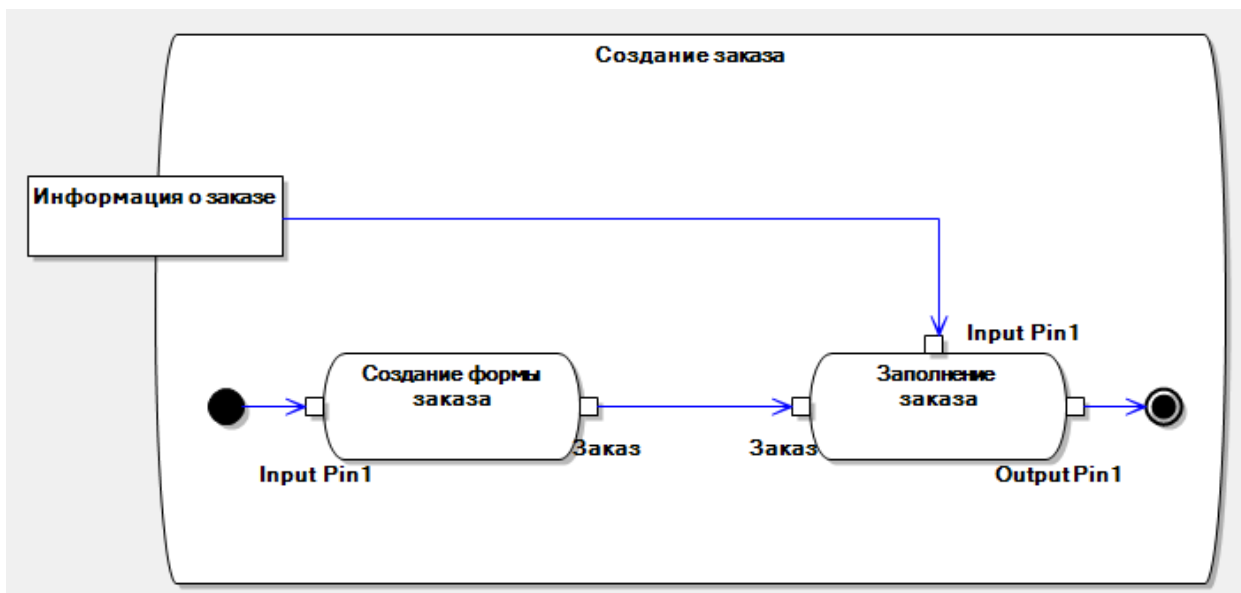


Рисунок 3 – Диаграмма активности, выполненная в Borland Together

Узлы управления в начале и в конце потока на рисунке 2 — это, соответственно, исходный и заключительный узлы. Когда вызывается деятельность “Создание заказа”, управляющий маркер помещается в начальный узел, а маркер данных с информацией о заказе — в объектный узел входного параметра. Управляющий маркер двигается от исходного узла к действию “Создание формы заказа”, которое начинает выполняться. Маркер данных передается от соответствующего параметра действию, вызывающему “Заполнение заказа”, которому приходится ждать до начала выполнения, пока “Создание формы заказа” не предоставит ему другие входные данные. После завершения действия “Заполнение заказа” управляющий маркер передается на конечный узел, деятельность завершается, а управление возвращается элементу, инициировавшему эту деятельность.

В случае моделирования операций диаграммы деятельности играют роль "продвинутых" блок-схем и применяются для подробного моделирования вычислений. На первое место при таком использовании выходят конструкции принятия решения, а также разделения и слияния

потоков управления (синхронизации). Этот способ применяется при детализации вариантов использования и других процедур.

Рекомендации по построению диаграмм деятельности для моделирования операций заключаются в следующем.

Процесс построения диаграммы деятельности можно описать в виде последовательности таких действий:

1) Составление перечня деятельностей в системе

Как исходные данные для этой операции хорошо подходит список вариантов использования (или список операций). Дополняться диаграммой деятельности может каждый сценарий использования. Можно также попытаться описать связь между ними.

2) Определение зависимостей между деятельностями

Для каждой деятельности нужно найти деятельности, непосредственно предшествующие (и следующие за ней тоже), то есть деятельности, без выполнения которых поток управления не может перейти к данной деятельности.

3) Выделение параллельных потоков деятельностей

Выделяются деятельности, имеющие общих предшественников.

4) Определение условий переходов

Для этого формулируются выражения, которые могут принимать только два значения - "истинно" или "ложно", соответствующие альтернативным потокам управления.

5) Уточнение сложных деятельностей

Повторяя пункты 1-4 для каждой из деятельностей (при необходимости), можно уточнить сложные деятельности.

Таким образом, диаграммы деятельности в UML используются для моделирования потоков различного типа: потоков сигналов или данных, а также алгоритмических или процедурных потоков.

Пример диаграммы деятельности, выполненной в MS Office Visio, представлен на рисунке 4.

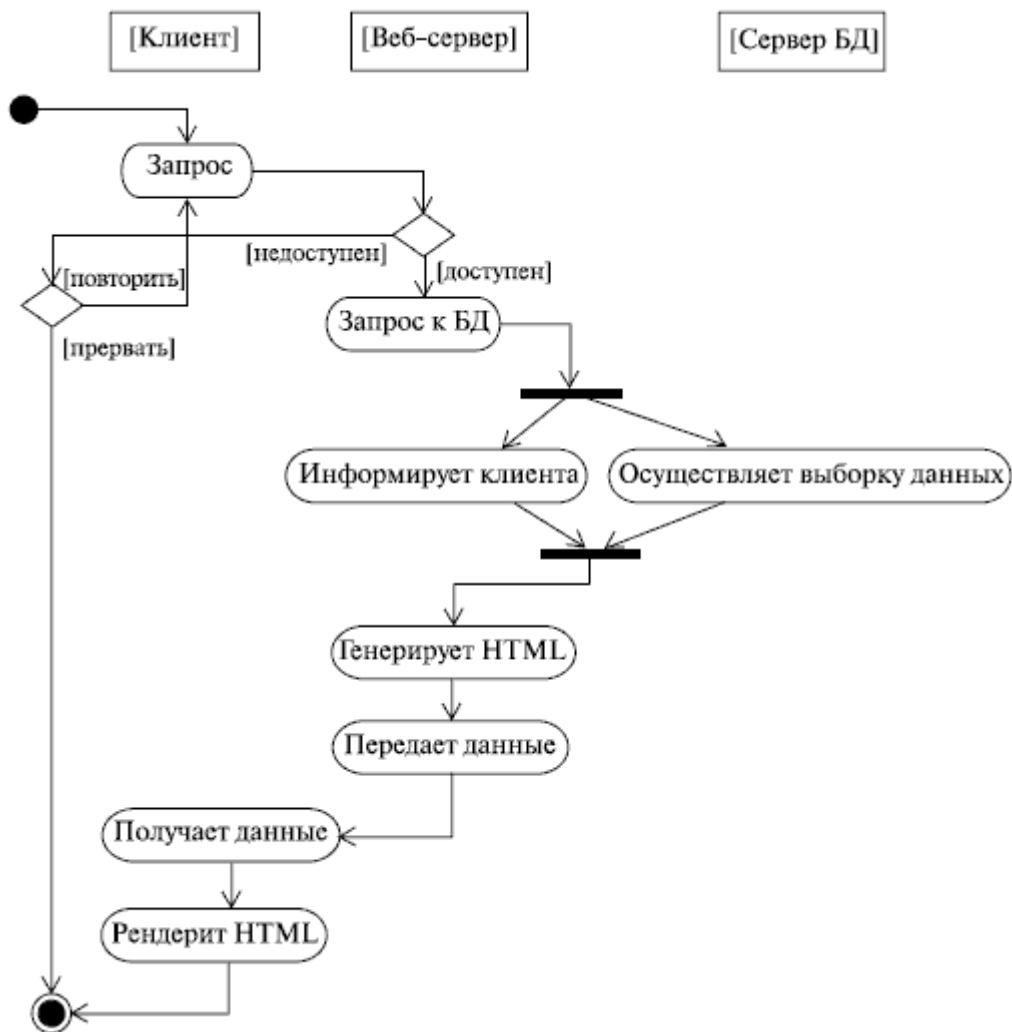


Рисунок 4 - Диаграмма деятельности, выполненная в MS Office Visio

На диаграмме показана работа с веб-приложением, решающим некую задачу в удаленной базе данных. Привлекает внимание расположение деятельностей на этой диаграмме: они как бы разбросаны по трем дорожкам, каждая из которых соответствует поведению одного из трех объектов - клиента, веб-сервера и сервера баз данных. Благодаря этому легко определить, каким из объектов выполняется каждая из активностей, что очень упрощает ее восприятие.

Аналогия с дорожками действительно очень удачна. Именно таково официальное название элемента нотации UML, позволяющего указать распределение ролей на диаграмме деятельности.

Создавая диаграммы деятельности, необходимо учитывать, что они лишь моделируют срез некоторых динамических аспектов поведения системы. С помощью единственной диаграммы деятельности никогда не удастся охватить все динамические аспекты системы. Вместо этого следует использовать разные диаграммы деятельности для моделирования динамики рабочих процессов или отдельных операций.

#### Спецификация требований к информационной системе

Основным средством спецификации требований к проектируемой информационной системе в рамках RUP является модель вариантов использования. Главное назначение диаграммы вариантов использования заключается в формализации функциональных требований к системе. Основная задача — представить единое средство, дающее возможность заказчику, конечному пользователю и разработчику совместно обсуждать функциональность и поведение системы.

Модель варианта использования дает подробную информацию о поведении системы или приложения, которое разрабатывается. Она определяет требования к системе в терминах требующейся функциональности (вариантов использования) для достижения целей или для решения проблемы, определенной пользователем. Она же описывает окружение(агенты) и отношения между вариантами использования и агентами. Модель вариантов использования обычно включает в себя диаграммы вариантов использования и диаграммы действий, которые описывают то, как пользователи общаются с системой.



Цель варианта использования заключается в том, чтобы определить законченный аспект или фрагмент поведения некоторой сущности без раскрытия внутренней структуры этой сущности. В качестве такой сущности может выступать исходная система или любой другой элемент модели, который обладает собственным поведением, подобно подсистеме или классу в модели системы.

Каждый вариант использования соответствует отдельному сервису, который предоставляет моделируемую сущность или систему по запросу пользователя (актера), т. е. определяет способ применения этой сущности. Сервис, который инициализируется по запросу пользователя, представляет собой законченную последовательность действий. Это означает, что после того как система закончит обработку запроса пользователя, она должна возвратиться в исходное состояние, в котором готова к выполнению следующих запросов.

Примерами вариантов использования могут являться следующие действия: проверка состояния текущего счета клиента, оформление заказа на покупку товара, получение дополнительной информации о кредитоспособности клиента, отображение графической формы на экране монитора и другие действия.

Пример простейшей диаграммы вариантов использования “Заказ товара”, выполненной в Rational Rose, представлен на рисунке 5. На диаграмме показаны условные графические изображения главных элементов диаграммы – действующего лица(актанта) и варианта использования, а также связи между ними.

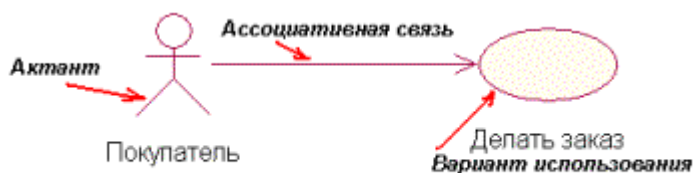


Рисунок 5 - Диаграмма вариантов использования,

выполненная в Rational Rose

Пример диаграммы вариантов использования, выполненной в Borland Together, показан на рисунке 6.

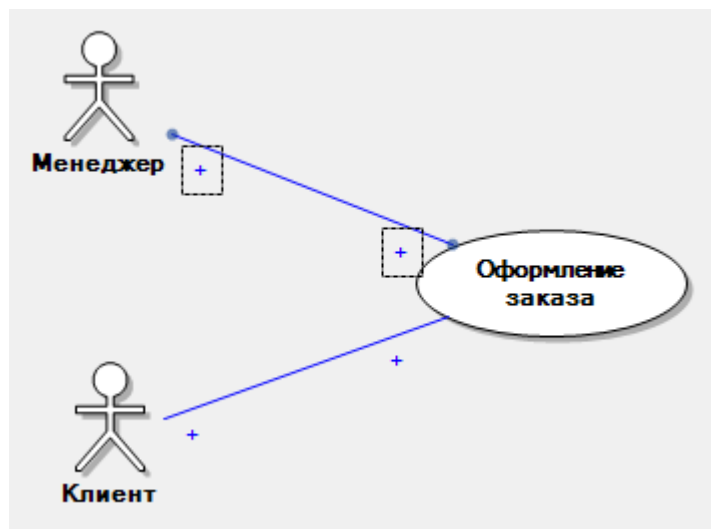


Рисунок 6 – Диаграмма вариантов использования, выполненная в Borland Together

Далее после создания диаграммы вариантов использования следует определить реализацию каждого варианта использования. Для этого применяются следующие способы:

- текстовое описание;
- описание алгоритма с помощью диаграмм деятельности;
- создание одной или несколько диаграмм взаимодействия.

Текстовое описание, соответствующее основной модели в рамках RUP, представляет собой:

- краткое описание;
- действующие лица;
- специальные требования;
- предпосылки;
- постусловия;
- точки расширения.

Например, рассмотрим вариант использования “Сделать предложение на аукционе”, диаграмма которого представлена на рисунке 7.

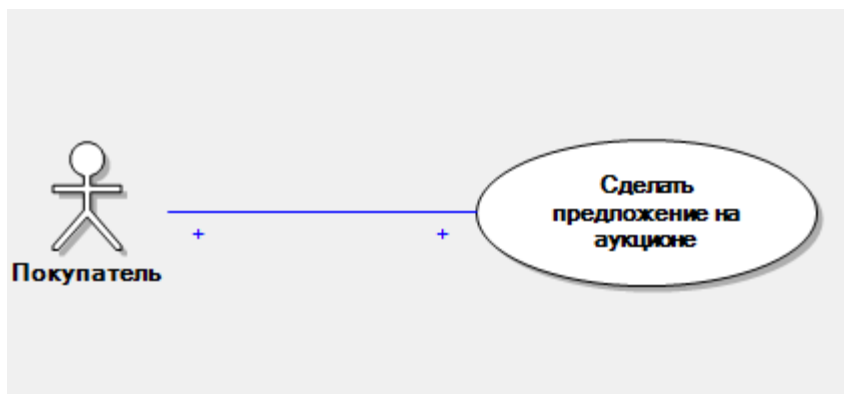


Рисунок 7 – Пример диаграммы варианта использования

В случае с приложением по ведению аукциона, речь может идти о представленном ниже потоке событий(последовательности, инициированной действующим лицом при подаче заявки в системе аукциона):

Основной поток:

-заявка (предложение цены): прецедент начинается в тот момент, когда покупатель предлагает свою цену на текущую позицию;

-ввод суммы: покупатель вводит сумму предложения. Система подтверждает, что сумма предложения превышает текущую ставку на значение, кратное шагу заявки для данной позиции;

-покупатель подтверждает заявку: покупатель подтверждает свое намерение разместить заявку;

-обработка заявки: система добавляет заявку к данной позиции;

-подтверждение заявки: система подтверждает наличие заявки путем отправки покупателю электронного сообщения. Продавец также уведомляется по электронной почте.

Альтернативные потоки операций могут описывать, что произойдет, если прием заявок будет прекращен до подачи предложения, если сумма заявки будет признана недействительной или покупатель не подтвердит подачу заявки.

Кроме текстового описания для детализации конкретного варианта использования(прецедента) можно построить диаграмму деятельности, правила построения которых аналогичны рассмотренных ранее.

Один из основных способов представления реализации варианта использования является создать одну или несколько диаграмм взаимодействия в форме диаграмм коммуникации или диаграмм последовательности, которые описывают один или несколько сценариев данного варианта использования. Этот способ в наибольшей степени соответствует идеологии UML и рекомендуется как основной и предпочтительный. Все эти операции выполняются на этапе анализа.

### **1.3.2 Анализ**

После определения контекста, в котором будет работать система и требований к ней, наступает черед анализа полученных данных. В процессе анализа создается **аналитическая модель(модель анализа)**, которая подводит разработчиков к архитектуре будущей системы. Аналитическая модель – это взгляд на систему изнутри, в отличие от модели вариантов использования, которая показывает, как система будет выглядеть снаружи.

Эта модель позволяет понять, как система должна быть спроектирована, какие в ней должны быть классы и как они должны взаимодействовать между собой. Основное ее назначение - определить направление реализации функциональности, выявленной на этапе сбора требований и сделать набросок архитектуры системы.

Модель анализа описывает логическую структуру системы и является фундаментом модели проектирования. Но в отличие от создаваемой в дальнейшем модели проектирования, модель анализа является в большей степени концептуальной моделью и только приближает разработчиков к классам реализации. Эта модель не должна иметь возможных противоречий, которые могут встретиться в модели вариантов использования.

Для построения аналитической модели выполняется анализ вариантов использования, который включает в себя:

- идентификацию классов, участвующих в реализации потоков событий;
- определение обязанностей классов;
- определение атрибутов и ассоциаций классов;
- унификацию классов анализа.

В потоках событий варианта использования выявляются классы трех типов:

- граничные классы, являющиеся посредниками при взаимодействии с внешними объектами;
- классы-сущности, представляющие собой основные абстракции (понятия) разрабатываемой системы;
- управляющие классы, обеспечивающие координацию поведения объектов в системе.

Классы анализа отражают функциональные требования к системе и моделируют объекты предметной области. Совокупность классов анализа представляет собой начальную концептуальную модель системы.

Для отображения модели анализа при помощи UML используется диаграмма классов со стереотипами (образцами поведения) «граничный класс», «сущность», «управление», а для детализации используются диаграммы взаимодействия, которые описывают взаимодействие групп объектов в различных условиях их поведения. Наиболее используемым типом таких диаграмм являются диаграммы коммуникации и последовательности.

#### Диаграмма коммуникации

Диаграмма коммуникации делает фокус на представлении группы взаимодействующих объектов и связей между ними, образующихся, если

объекты общаются друг с другом посредством отсылки и приема сообщений. Также диаграммы коммуникаций подобны диаграммам объектов, но на них дополнительно могут быть показаны отсылаемые сообщения, причем допускается даже с указанием нумерации, описывающей порядок их следования во времени.

Диаграмма коммуникации используется для описания поведения системы как последовательности обмена сообщениями между элементами.

Основные сущности, используемые на диаграмме:

- роли, которые играют взаимодействующие элементы;
- объекты – экземпляры конкретных классов;
- связи - отношения, соединяющие взаимодействующие элементы.

Диаграмма коммуникации описывает поведение как взаимодействие, т. е. как протокол обмена сообщений между объектами.

Построение диаграммы коммуникации (кооперации) можно начинать сразу после построения диаграммы вариантов использования. В этом случае каждый из вариантов использования может быть специфицирован в виде отдельной диаграммы кооперации уровня спецификации.

Главная особенность диаграммы коммуникации (кооперации, сотрудничества) заключается в возможности графически представить не только последовательность взаимодействия, но и все структурные отношения между объектами, участвующими в этом взаимодействии.

Прежде всего, на диаграмме коммуникации(кооперации, сотрудничества) в виде прямоугольников(окружностей) изображаются участвующие во взаимодействии объекты, содержащие имя объекта, его класс и, возможно, значения атрибутов. Далее должны быть изображены динамические связи - потоки сообщений. Они представляются в виде соединительных линий между объектами, над которыми располагается стрелка с указанием направления, имени сообщения и порядкового номера в общей последовательности инициализации сообщений.

Пример диаграммы коммуникации (сотрудничества) показан на рисунке 8. Здесь показаны объект класса сущности “счет”, объект класса управления “обработчик” и объект граничного класса “интерфейс запроса на оплату”.

Стереотип «граничный класс» отображает класс, который взаимодействует с внешними актантами, «сущность» – отображает классы, которые являются хранилищами данных, а «управление» – классы, управляющие запросами к сущностям.

Линии, соединяющие объекты классов, отражают их взаимодействие.

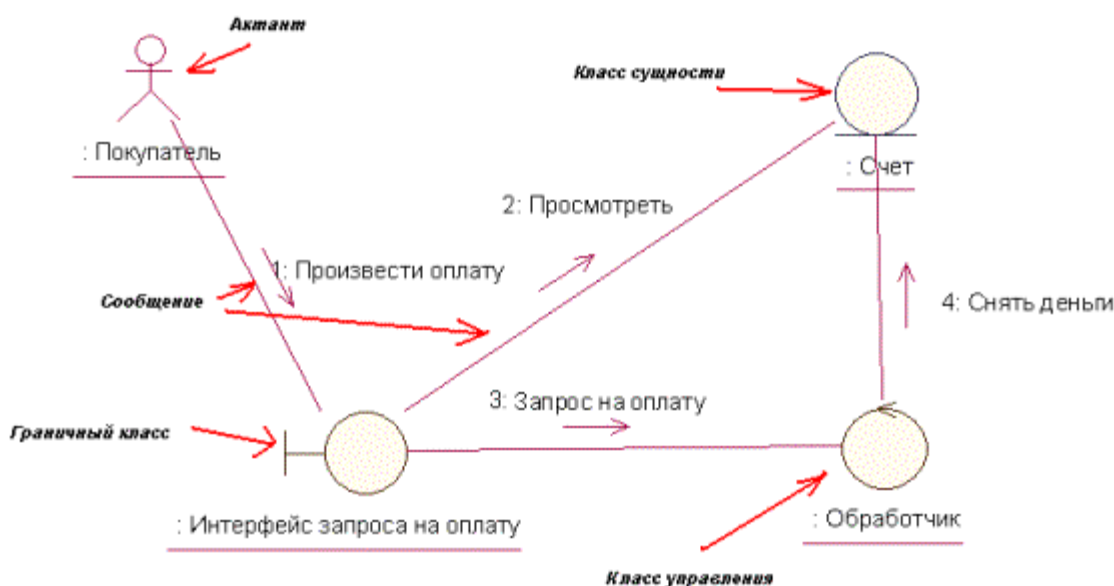


Рисунок 8 - Пример диаграммы коммуникации (сотрудничества)

Нумерация сообщений показывает их порядок, однако назначение диаграммы не в том, чтобы рассмотреть порядок обмена сообщениями, а в том, чтобы наглядно показать связи классов друг с другом.

При создании диаграммы коммуникации можно явно указать имена ассоциаций и ролей, которые играют объекты в данной ассоциации, как показано на диаграмме, изображенной на рисунке 9. Здесь показаны ассоциации “Продажа товара” и “Продажа компьютера”, а также роли “клиент” и “менеджер”.



Рисунок 9 - Пример изображения ассоциаций и ролей

### Диаграмма последовательности

Если акцентировать внимание на порядке взаимодействия, то другим его представлением будет диаграмма последовательности (Sequence).

Диаграмма последовательности - это диаграмма, чаще всего, описывающая один сценарий приложения. На диаграмме изображаются экземпляры объектов и сообщения, которыми они обмениваются в рамках одного варианта использования. Участники диаграммы именуется следующим образом: **имя: Класс**, где и имя, и класс являются не обязательными, но если используется класс, то присутствие двоеточия обязательно.

На диаграмме последовательности, каждый участник представлен вместе со своей линией жизни (lifeline), это вертикальная линия под объектом, вертикально упорядочивающая сообщения на странице. Важно: все сообщения на диаграмме следует читать сверху вниз. Каждая линия жизни имеет полосу активности (прямоугольники), которая показывает интервал активности каждого участника при взаимодействии.

Обозначение различных сообщений на диаграмме показано на рисунке 10.





Рисунок 10 - Обозначение различных сообщений на диаграмме

У первого сообщения нет участника, пославшего его, поскольку оно приходит от неизвестного источника. Такое сообщение называется найденным сообщением (found message). Отправитель или получатель сообщения может находиться за пределами диаграммы коммуникации, и в этом случае используют входной и выходной шлюзы.

Сообщения, которыми обмениваются элементы, могут быть синхронными или асинхронными, что отражается в нотации стрелочек. Синхронные (synchronous message) - требующие возврата ответа, а асинхронные (asynchronous message) – ответа не требуют (вызывающий объект может продолжать работу). На диаграмме синхронные вызовы обозначаются закрашенными стрелочками, асинхронные – не закрашенными или половинными стрелочками.

Обратной пунктирной стрелкой показывается возврат ответа на сообщение (если сообщение является синхронным). Лучше применять изображение возврата только в тех случаях, когда это поможет лучше понять устройство взаимодействия. Во всех остальных случаях, стоит опускать изображения возвратов, т.к. они будут вносить некоторую неразбериху. Просто, при использовании синхронного сообщения, стоит помнить, что у него всегда есть возврат.

Если элемент диаграммы связан сам с собой, то такая связь называется рефлексивной (самовывоз).

Если в сообщении требуется передать параметры, то они указываются в скобках через запятую, с указанием типа параметра (messageText(text : string)).

Для того чтобы задать порядок следования сообщений, используют десятичную нумерацию.

Пример диаграммы последовательности показан на рисунке 11. На ней представлены два объекта и все возможные виды сообщений.

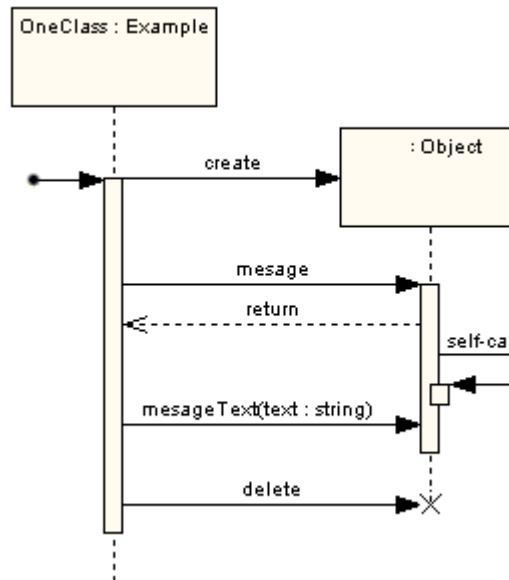


Рисунок 11 - Виды сообщений, которыми обмениваются объекты

Диаграмма последовательности, соответствующая диаграмме сотрудничества, показанной на рисунке 8, представлена на рисунке 12. Виды сообщений, которыми обмениваются объекты, аналогичны сообщениям диаграммы коммуникации, изображенным на рисунке 10.

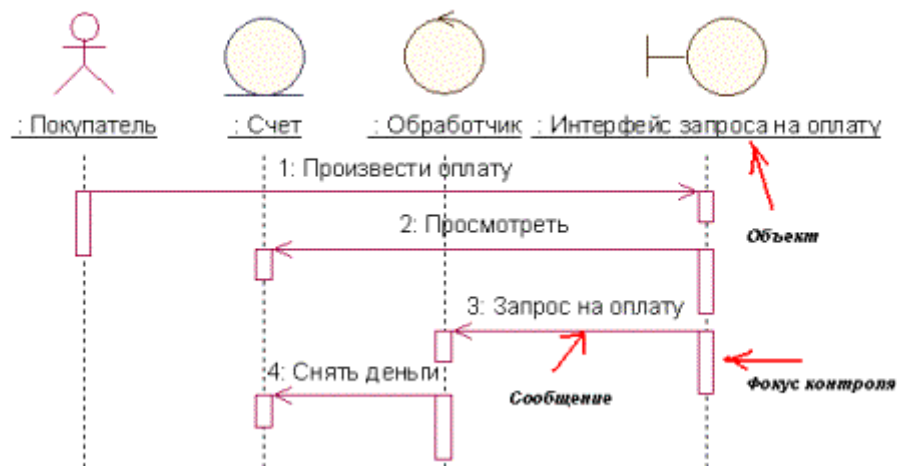


Рисунок 12 - Пример диаграммы последовательности действий

Решение о том какую из двух диаграмм нужно создавать первой, зависит от предпочтений конкретного разработчика. Поскольку эти диаграммы являются отображением одного и того же процесса, то и та и другая позволяют отразить взаимодействие между объектами.

Диаграммы коммуникации и последовательности транзитивны, выражают взаимодействие, но показывают его различными способами и с достаточной степенью точности могут быть преобразованы одна в другую.

При использовании такого инструмента для создания моделей как Rational Rose, эти два вида диаграмм могут быть созданы друг из друга автоматически [5].

В некоторых случаях могут строиться диаграммы обзора взаимодействия и диаграммы синхронизации.

### 1.3.3 Проектирование

Следующим этапом в процессе создания системы будет проектирование, в ходе которого на основании моделей, созданных ранее, создается **модель проектирования**. Эта модель отражает физическую реализации системы и описывает создаваемый продукт на уровне классов и компонентов. В отличие от модели анализа, модель проектирования имеет явно выраженную зависимость от условий реализации, применяемых языков программирования и компонентов.

**Модель проектирования**, используя различные диаграммы (в том числе диаграммы последовательности, машины состояний, компонента и размещения), подробно описывает, как устроено приложение и как оно будет реализовываться. Она также описывает структурные компоненты программ и технологий, например, обеспечивающих **персистентность**, распределение, безопасность и доступ к данным.

Для максимально точного понимания архитектуры системы, эта модель должна быть максимально формализована, и поддерживаться в актуальном состоянии на протяжении всего жизненного цикла разработки системы.

В RUP проектирование концентрируется вокруг определения архитектуры системы, а для систем с большой долей программного обеспечения - вокруг архитектуры программного обеспечения. Использование компонентных архитектур - один из шести наилучших подходов к разработке программ, инкорпорированных в RUP, рекомендует уделять больше времени на разработку и сопровождение архитектур. Время, затраченное на эти усилия, сокращает риски, связанные с ненадежными и негибкими системами.

Для создания модели проектирования используются целый набор UML диаграмм: диаграммы классов, диаграммы композитной структуры(кооперации), диаграммы взаимодействия, диаграммы активности. Основной является диаграмма классов.

### Диаграмма классов

Диаграмма классов является основным типом диаграммы статической структуры. Она описывает структуру системы, показывая её классы, их атрибуты и операторы, и также взаимосвязи этих классов.

Каждый класс имеет имя, размещенное в верхнем блоке прямоугольника, изображающего класс. Для атрибутов и операций в элементах отводится отдельный блок. Каждый блок разделяется горизонтальной чертой.

Для атрибутов и операций применяются спецификаторы доступа. Спецификатора доступа языка C++ (public, private, protected) в UML отображаются символами + (public), - (private), # (protected), которые ставятся

перед именем атрибута/операции. Также возможен вариант с ключевыми словами `public`, `private`, `protected`. Значение спецификаторов доступа: `public` - поля/методы класса видны снаружи класса. Т.е. к ним могут получать доступ объекты класса. `private` - поля/методы класса видны только внутри определения класса. `protected` - поля/методы класса видны в определении самого класса и в определениях производных классов.

Между классами существуют различные виды взаимодействия (или связи): один класс может быть производным другого, третий может содержать объект четвертого в виде поля и т.д. Для различных видов взаимодействия в UML есть специальные названия.

Первый вид взаимодействия - ассоциация(association).

**Ассоциация** – это семейство связей двух и более классов. Обычно ассоциация возникает, когда один класс вызывает метод другого или если при вызове метода в качестве аргумента передается объект другого класса. Иногда при ассоциации показывают направленность (если это имеет значение).

Частным случаем ассоциации является связь – простая взаимосвязь между объектами. Она представляется линией соединяющей два или более объектных блока. Она встречается на диаграммах классов или объектов.

Всего существует пять типов ассоциации. Но наиболее распространены два: двунаправленная и однонаправленная ассоциации.

Сообщение направленная ассоциация(Message/Directed Association) используется, когда один класс “общается” с другим при создании экземпляра класса. Экземпляр класса — это описание конкретного объекта в памяти. Класс описывает свойства и методы, которые будут доступны у объекта, построенного по описанию, заложенному в класс. Экземпляры используют для представления конкретных сущностей реального мира.

Графически направленная ассоциация представляется в виде стрелочки направленной к “вызываемому” классу.

Частными вариантами ассоциации являются: агрегация и композиция.

**Агрегация**(*быть частью*) применяется, когда один класс должен быть контейнером других классов. Причем время существования содержащихся классов никак не зависит от времени существования класса контейнера. Графически агрегация представляется пустым ромбиком на блоке класса и линией, идущей от этого ромбика к содержащемуся классу.

**Композиция** - еще один случай ассоциации, но более строгий. В отличие от агрегации, композиция имеет жёсткую зависимость времени существования экземпляров класса контейнера и экземпляров содержащихся классов. Если контейнер будет уничтожен, то всё его содержимое будет уничтожено также. Графически представляется, как и агрегация, но с закрашенным ромбиком.

Различие между этими двумя видами ассоциации состоит в том, что композиция может быть частью одного и только одного целого, в то время как агрегация может быть частью нескольких объектов.

Еще одной разновидностью связи является **генерализация** (обобщение). Генерализация показывает, что один из двух связанных классов (*подтип*), является более частной формой другого (*супертип*), который называется обобщением первого. Графически генерализация представляется линией с пустым треугольником у супертипа.

Последнее отношение, которое мы рассмотрим, будет **реализация**(*realization*). Данная связь показывает отношение: класс - объект. На диаграмме реализация показывается пунктирной линией и не закрашенной стрелочкой.

Одной из важнейших характеристик взаимодействия является кратность(*multiplicity*) роли ассоциации. Кратностью роли ассоциации называется характеристика, указывающая, сколько объектов класса с данной ролью может или должно участвовать в каждом экземпляре ассоциации.

Наиболее распространенным способом задания кратности роли ассоциации является указание конкретного числа или диапазона. Например, указание “1” говорит о том, что все объекты класса с данной ролью должны участвовать в некотором экземпляре данной ассоциации, причем в каждом экземпляре ассоциации может участвовать ровно один объект класса с данной ролью. Указание диапазона “0..1” говорит о том, что не все объекты класса с данной ролью обязаны участвовать в каком-либо экземпляре данной ассоциации, но в каждом экземпляре ассоциации может участвовать только один объект. Аналогично, указание диапазона “1..\*” говорит, что все объекты класса с данной ролью должны участвовать в некотором экземпляре данной ассоциации, и в каждом экземпляре ассоциации должен участвовать хотя бы один объект (верхняя граница не задана).

Пример диаграммы классов, на которой показаны все возможные варианты связей, представлен на рисунке 13.

Проектирование классов на данном этапе включает следующие действия:

- детализация проектных классов;
- уточнение операций и атрибутов;
- уточнение связей между классами;
- моделирование состояний для классов.

**Детализация проектных классов**, определенных в процессе анализа, уточнение атрибутов классов заключается в следующем:

- задается тип атрибута и значение по умолчанию (необязательно);
- задается видимость атрибутов: public, private или protected;
- при необходимости определяются производные (вычисляемые) атрибуты.

Обязанности классов, определенные в процессе анализа и документированные в виде «операций анализа», преобразуются в операции, которые будут реализованы в коде. При этом:

- каждой операции присваивается краткое имя, характеризующее ее результат;
- определяется полная сигнатура операции;
- создается краткое описание операции, включая смысл всех ее параметров;
- определяется видимость операции: public, private или protected;
- определяется область действия операции: операция объекта или операция класса.

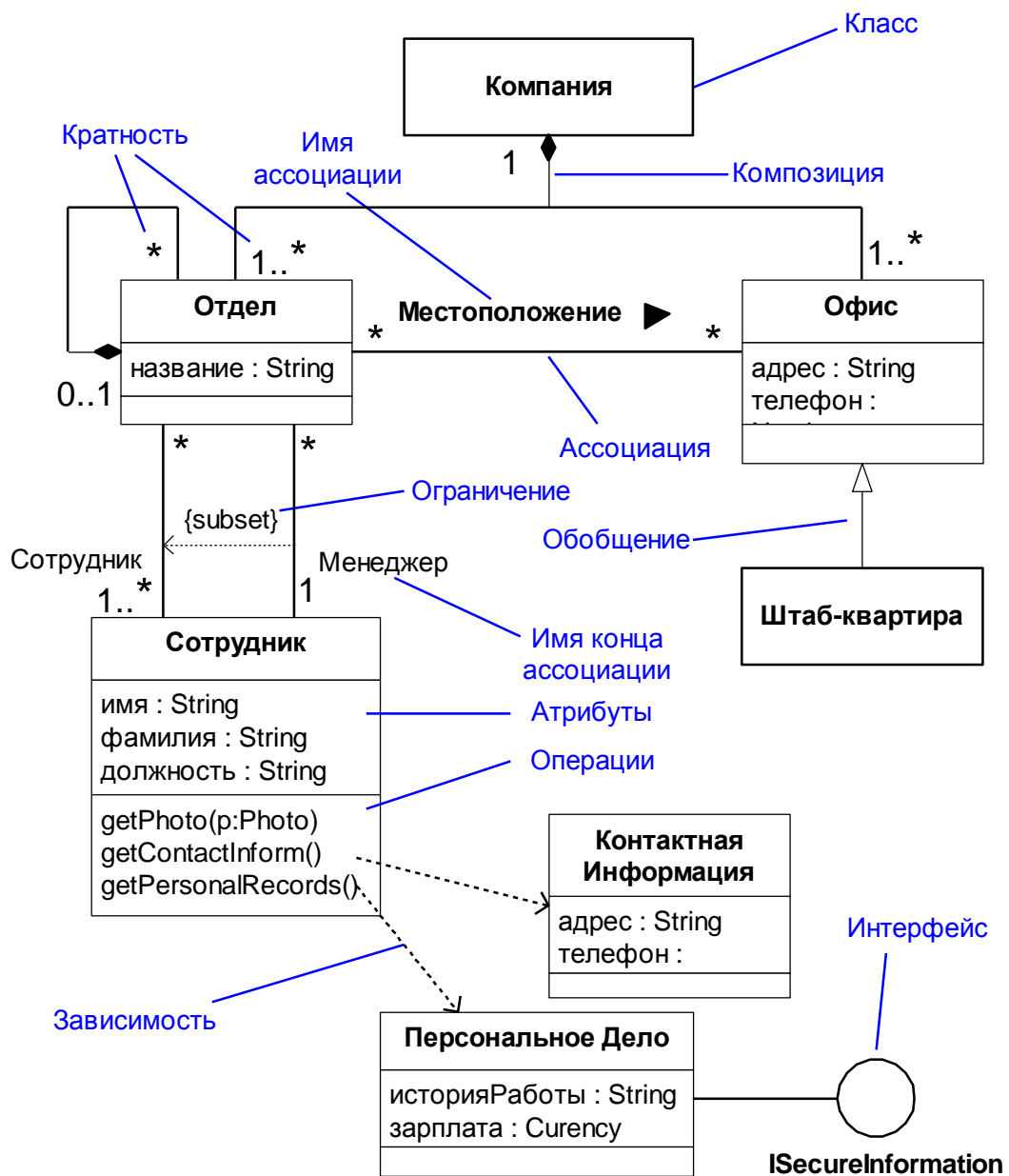


Рисунок 13 - Пример диаграммы классов



В процессе проектирования связи между классами подлежат уточнению. Ассоциации между граничными и управляющими классами преобразуются в зависимости. Агрегации, обладающие свойствами композиции, преобразуются в связи композиции. Связи обобщения могут преобразовываться в ситуациях с так называемой метаморфозой подтипов, когда объект суперкласса может менять свой подтип

Например, для бизнес-процесса – продажа товаров по заказу(каталогу), можно выделить классы Заказ и Клиент, который может представляться как юридическим, так и физическим лицом. Каждый класс имеет определенные атрибуты(свойства). Заказ выполняется в определенный день. Клиент имеет имя и т.д.

Пример диаграммы классов, созданной в среде Rational Rose, показан на рисунке 14.

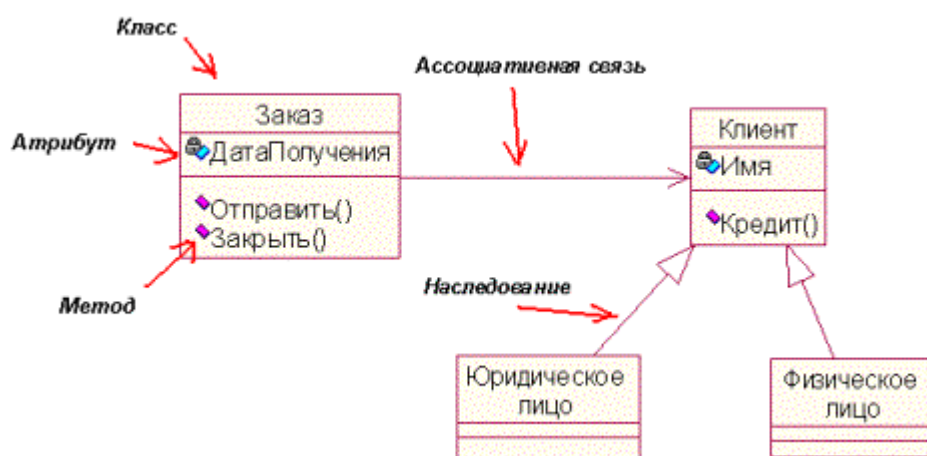


Рисунок 14 - Пример диаграммы классов, созданной в среде Rational Rose

Из диаграммы видно, что между классами “Заказ” и “Клиент” имеет место связь однонаправленная ассоциация.

Из диаграммы видно, что базовый класс “Клиент” имеет два производных класса “Юридическое лицо” и “Физическое лицо”,

соединенных с базовым классом связью типа “Обобщение”, реализующей принцип наследования.

Другой вариант диаграммы классов для того же бизнес-процесса, созданной в среде Borland Together, показан на рисунке 15. На этой диаграмме классы имеют более полный набор атрибутов и операций. Однако класса Клиент не является базовым. Наследования классов на данной диаграмме не предусмотрено.

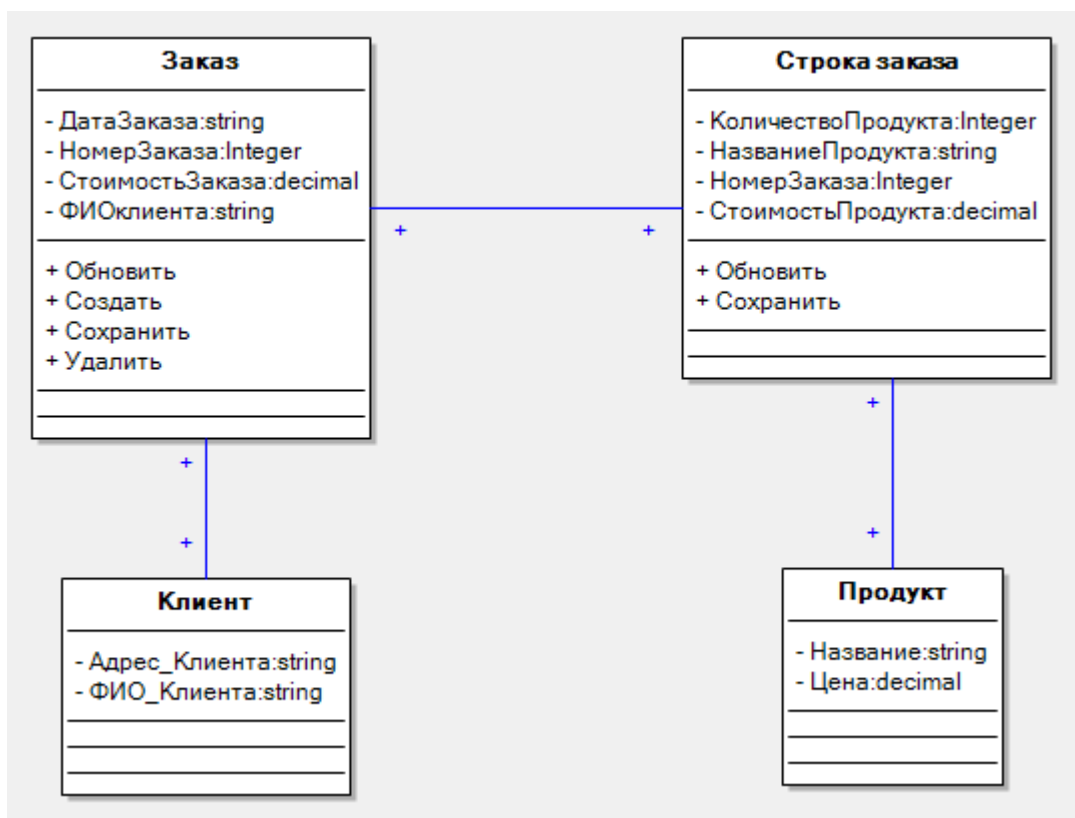


Рисунок 15 - Диаграмма классов, созданная в среде Borland Together

### Основные правила построения диаграмм классов

В UML необязательно расписывать все детали классов. Это будет сделано при написании кода на конкретном языке (в нашем случае - C++). В UML-диаграмме можно опускать ненужные детали. Например, в диаграмму элемента можно добавить только те операции/атрибуты, которые важны для данной диаграммы, неважные особенности класса в UML можно опускать.

Для более полного раскрытия архитектуры проектируемой системы могут строиться диаграммы композитной/составной структуры и диаграммы автомата.

#### Диаграмма композитной/составной структуры

**Диаграмма композитной/составной структуры** — статическая структурная диаграмма, демонстрирует внутреннюю структуру классов и, по возможности, взаимодействие элементов (частей) внутренней структуры класса.

Подвидом диаграмм композитной структуры являются диаграммы кооперации (Collaboration diagram, введены в UML 2.0), которые показывают роли и взаимодействие классов в рамках кооперации. Кооперации удобны при моделировании шаблонов проектирования.

Диаграммы композитной структуры могут использоваться совместно с диаграммами классов, как показано на рисунке 16.

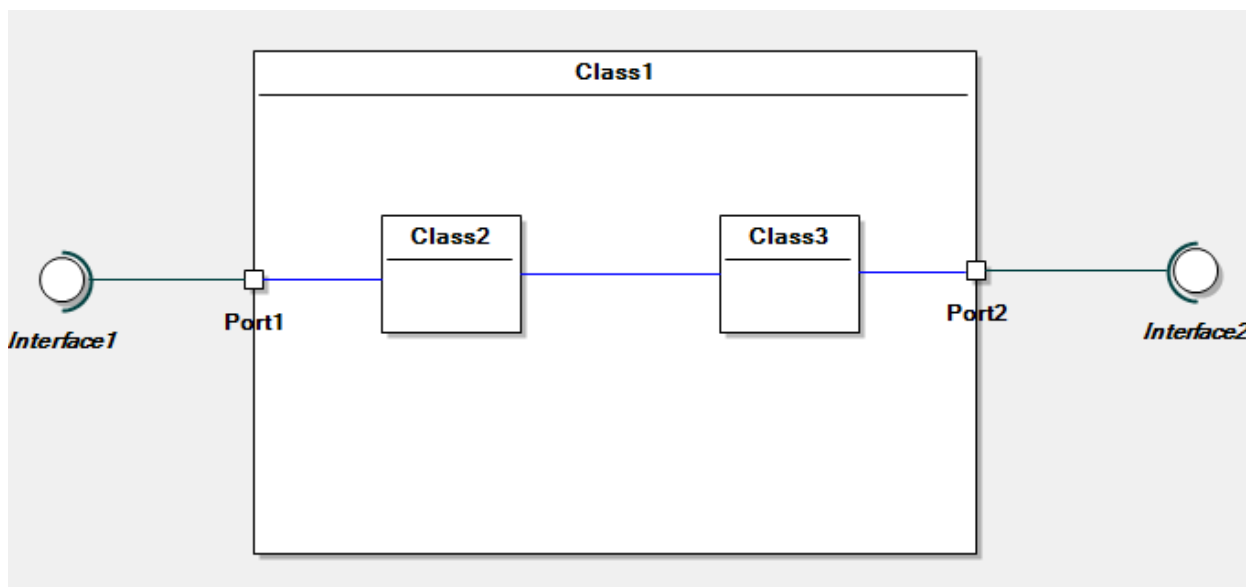


Рисунок 16 - Диаграмма композитной структуры, созданная в среде Borland Together

## Диаграммы автомата

Диаграмма автомата, State Machine diagram (диаграмма конечного автомата, диаграмма состояний) — диаграмма, на которой представлен конечный автомат с простыми состояниями, переходами и композитными состояниями, как показано на рисунке 17.

Конечный автомат (State machine) — спецификация последовательности состояний, через которые проходит объект или взаимодействие в ответ на события своей жизни, а также ответные действия объекта на эти события. Конечный автомат прикреплен к исходному элементу (классу, кооперации или методу) и служит для определения поведения его экземпляров.

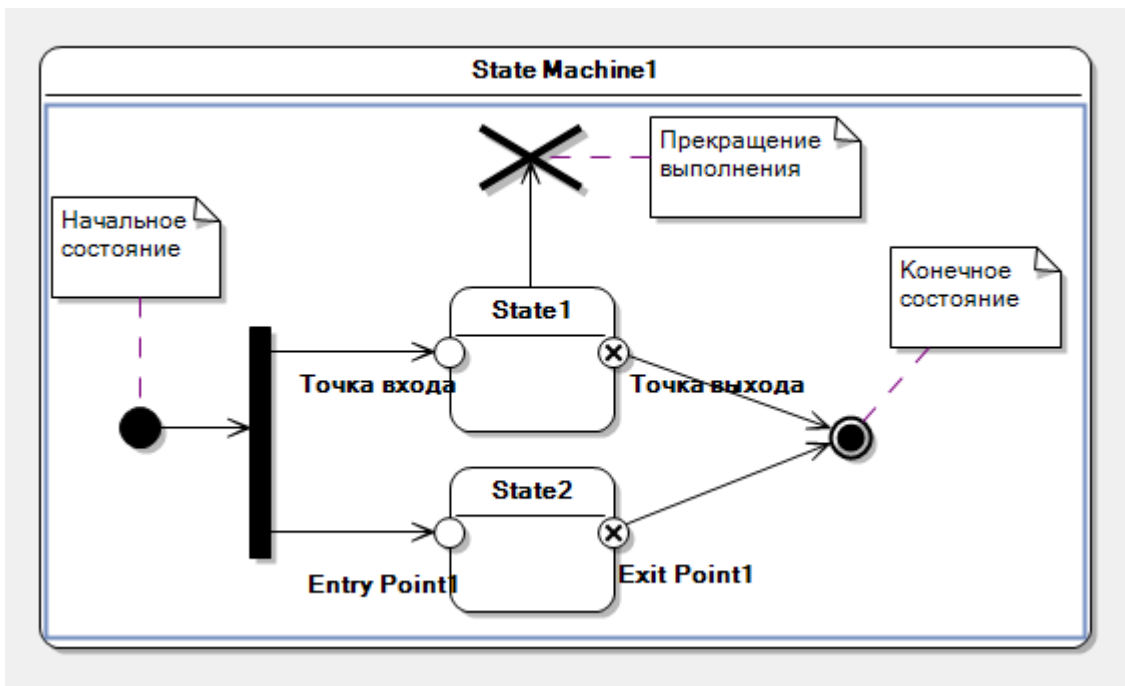


Рисунок 17 – Диаграмма автомата, созданная в среде Borland Together

Если в системе присутствуют объекты со сложным поведением, то строят диаграммы состояний. Построение диаграмм состояний может оказать следующее воздействие на описание классов:

- события могут отображаться в операции класса;

-особенности конкретных состояний могут повлиять на детали выполнения операций;

-описание состояний и переходов может помочь при определении атрибутов класса.

Каждая диаграмма состояний в UML описывает все возможные состояния одного экземпляра определенного класса и возможные последовательности его переходов из одного состояния в другое, то есть моделирует все изменения состояний объекта как его реакцию на внешние воздействия.

Диаграммы состояний чаще всего используются для описания поведения отдельных объектов, но также могут быть применены для спецификации функциональности других компонентов моделей, таких как варианты использования, актеры, подсистемы, операции и методы.

Главное предназначение этой диаграммы — описать возможные последовательности состояний и переходов, которые в совокупности характеризуют поведение элемента модели в течение его жизненного цикла.

Действие (action), как уже говорилось, является непрерываемым поведением, осуществляющимся как часть перехода. Входные и выходные действия показывают внутри состояний, поскольку они определяют, что происходит, когда объект входит или выходит из состояния. Большую часть действий, однако, изображают вдоль линии перехода, так как они не должны осуществляться при входе или выходе из состояния.

Действие рисуют вдоль линии перехода после имени события, его изображению предшествует наклонная (косая) черта.

Событие или действие может быть поведением внутри объекта, а может представлять собой сообщение, посылаемое другому объекту. Если событие или действие посылается другому объекту, перед ним на диаграмме помещают знак «^».

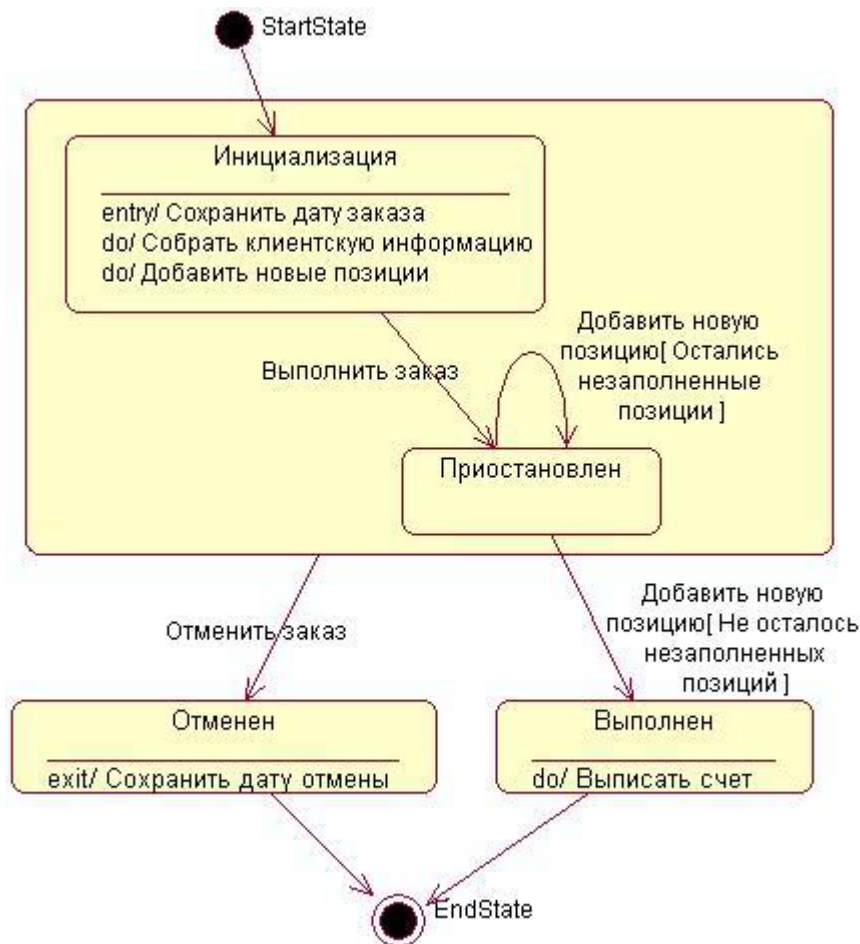


Рисунок 18 – Диаграмма состояний, созданная в среде Rational Rose

Для группировки классов, обладающих некоторой общностью, применяются пакеты. Пакет – общий механизм для организации элементов модели в группы. Каждый пакет – это группа элементов модели, иногда сопровождаемая диаграммами, поясняющими структуру группы. Каждый элемент модели может входить только в один пакет. *Диаграммы пакетов* отображают зависимости между пакетами, возникающие, если элемент одного пакета зависит от элемента другого.

Пакеты также используются для представления подсистем. Подсистема – это комбинация пакета (поскольку она включает некоторое множество классов) и класса (поскольку она обладает поведением, т.е. реализует набор операций, которые определены в ее интерфейсах). Связь между подсистемой и интерфейсом называется связью реализации.

Жёсткого разделения между разными структурными диаграммами не проводится, поэтому данное название предлагается исключительно для удобства и не имеет семантического значения (пакеты и диаграммы пакетов могут присутствовать на других структурных диаграммах).

### **1.3.4 Реализация**

Основная задача процесса реализации – создание системы в виде компонентов – исходных текстов программ, сценариев, двоичных файлов, исполняемых модулей и т.д. На этом этапе создается модель реализации, которая описывает то, как реализуются элементы модели проектирования, какие классы будут включены в конкретные компоненты. Данная модель описывает способ организации этих компонентов в соответствии с механизмами структурирования и разбиения на модули, принятыми в выбранной среде программирования и представляется диаграммой компонентов

#### **Диаграмма компонентов**

Диаграмма компонентов, в отличие от ранее рассмотренных диаграмм, описывает особенности физического представления системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код. В качестве физических компонент могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и т. п.

Общий вид структуры информационной системы в виде диаграммы компонентов показан на рисунках 19 и 20. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними. Пунктирные стрелки, соединяющие модули,

показывают отношения взаимозависимости, аналогичные тем, которые имеют место при компиляции исходных текстов программ или реализации интерфейсов.

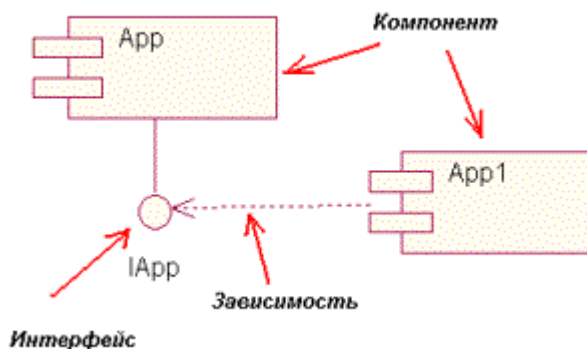


Рисунок 19 - Пример диаграммы компонентов на UML 1.5, созданной в среде Rational Rose

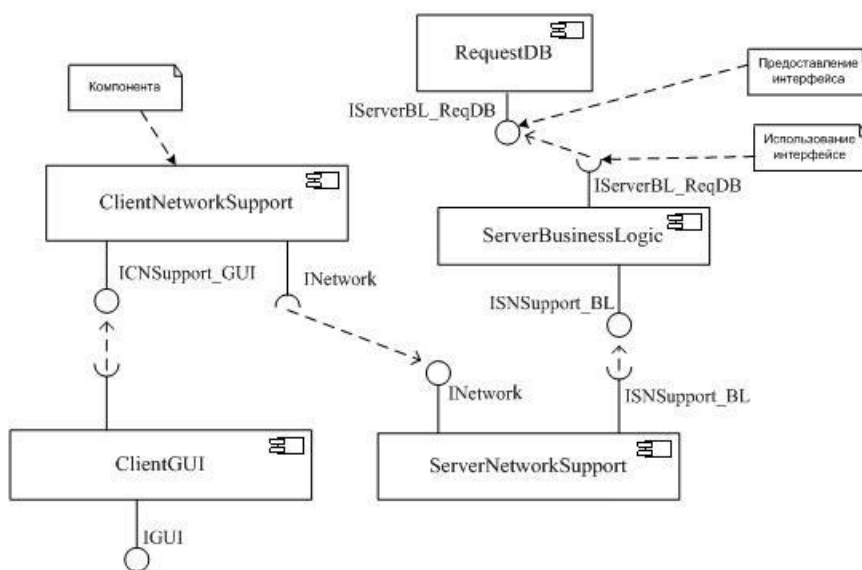


Рисунок 19 - Пример диаграммы компонентов на UML 2.0, созданной в среде Borland Together

### Диаграмма развертывания

Физическое представление программной системы не может быть полным, если отсутствует информация о том, на какой платформе и на каких



вычислительных средствах она реализована. Если разрабатывается программа, выполняющаяся локально на компьютере пользователя и не использующая периферийных устройств и ресурсов, то в разработке дополнительных диаграмм нет необходимости. При разработке же корпоративных приложений наличие таких диаграмм может быть крайне полезным для решения задач рационального размещения компонентов в целях эффективного использования распределенных вычислительных и коммуникационных ресурсов сети, обеспечения безопасности и других.

Для представления общей конфигурации и топологии распределенной программной системы в UML предназначены диаграммы развертывания.

Диаграмма развёртывания, *Deployment diagram* — служит для моделирования работающих узлов (аппаратных средств) и артефактов, развёрнутых на них. В UML 2 на узлах разворачиваются артефакты (*artifact*), в то время как в UML 1 на узлах разворачивались компоненты. Между артефактом и логическим элементом (компонентом), который он реализует, устанавливается зависимость манифестации. Это самый простой тип диаграмм, предназначенный для моделирования распределения устройств в сети. Для отображения используется всего два варианта значков процессор и устройство вместе со связями между ними.

Разработка диаграммы развертывания начинается с идентификации всех аппаратных, механических и других типов устройств, которые необходимы для выполнения системой всех своих функций. В первую очередь специфицируются вычислительные узлы системы, обладающие памятью и/или процессором.

Один из возможных вариантов построения диаграммы развертывания, созданной в среде *Borland Together*, показан на рисунке 21.

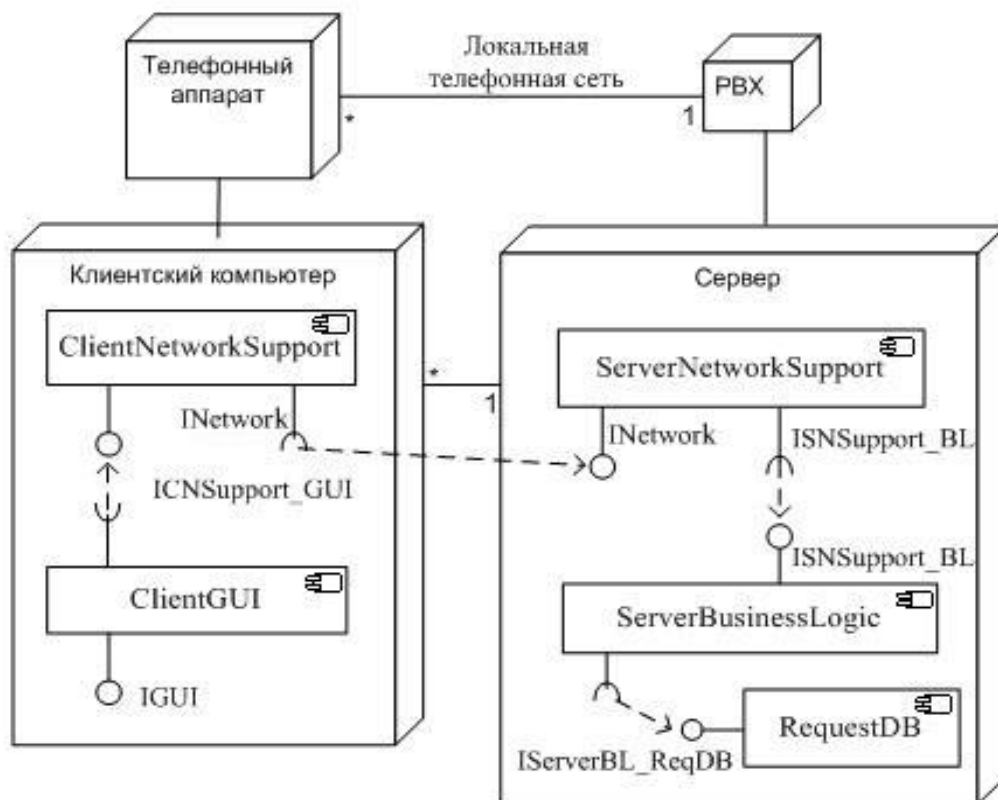


Рисунок 21 - Диаграмма развертывания, созданная в среде Borland Together

На диаграмме, показано каким образом компоненты телефонной службы приема заявок распределяются по аппаратной части системы.

## 2 Методические рекомендации по выполнению курсового проекта

### 2.1 Общие положения

Целью курсового проекта является закрепление теоретических знаний по методам проектирования информационных систем, а также приобретение практических навыков объектно-ориентированного анализа и проектирования информационных систем в современной среде разработки. Для создания диаграмм можно использовать любое современное CASE-средство, например IBM Rational Rose, Borland Together, MS Visio и др.

Тематика курсовых проектов уточняется на кафедре перед началом семестра, утверждается заведующим кафедрой и выдается студентам в начале семестра, в котором выполняется курсовой проект.

Студенты обязаны:

-выполнить и оформить пояснительную записку (ПЗ) к курсовому проекту в соответствии с требованиями данных методических указаний, а также **Правил оформления и требований к содержанию курсовых проектов (работ) и выпускных квалификационных работ**, разработанных в университете [4];

-представить ПЗ на проверку преподавателю (зарегистрировать в лаборатории кафедры) не позднее, чем за неделю до начала очередной сессии;

-защитить допущенную преподавателем курсовую работу перед членами комиссии не позднее, чем за три дня до начала очередной сессии.

Студенты, не выполнившие курсовую работу по дисциплине, не защитившие ее в установленные сроки или получившие за нее неудовлетворительную оценку, к сессии не допускаются.

Курсовой проект выполняется в часы самостоятельной работы лично студентом. За все принятые решения и используемые фактические данные, за правильность всех выводов и рекомендаций отвечает студент.

Порядок выполнения курсового проекта предусматривает:

-получение задания на курсовую работу, уяснение темы курсового проекта и требований по ее выполнению;

-изучение методики построения диаграмм;

-получение консультации в ходе выполнения работы;

-построение моделей информационной системы согласно заданию;

-оформление курсового проекта, курсовую работу необходимо распечатать на компьютере и сшить пластиковым скоросшивателем;

-представления курсового проекта на контроль руководителю;

-защита курсового проекта.

## 2.2 Содержание курсового проекта

Курсовой проект состоит из следующей документации:

- пояснительная записка;
- графическая часть (плакаты, комплект конструкторской документации).

Пояснительная записка курсового проекта должна в краткой и четкой форме раскрывать постановку задачи, выбор и обоснование принципиальных решений, содержать описание используемых методов и средств, анализ результатов работы и выводы.

Пояснительная записка включает в себя текстовый материал, диаграммы, схемы моделей, спроектированные в ходе самостоятельной работы над курсовой работой и должна быть объемом 25÷55 листов текста.

Материал в пояснительной записке размещают следующим образом:

- титульный лист;
- задание;
- содержание;
- введение;
- основная часть, включающая изложение разделов, предусмотренных заданием;
- заключение;
- список использованных источников;
- ведомость проекта;
- приложения (если имеются в наличии).

Графическая часть включает различные диаграммы и другие документы, предусмотренные заданием. Объем графической части определяет руководитель курсового проекта согласно теме работы и

методическим рекомендациям по выполнению работы. Иллюстративный материал может быть выполнен в виде раздаточного материала, плакатов или электронной презентации на компьютере (в виде слайдов).

Титульные листы курсового проекта должны быть изготовлены с применением печатающих устройств вывода ЭВМ. Образец титульного листа берется на кафедре.

Задания на курсовые работы) оформляются на бланках, изготовленных с применением печатающих устройств вывода ЭВМ, и снабжаются соответствующими подписями. Оформленные бланки задания выдаются студентам до начала выполнения работы.

В пояснительной записке помещают содержание, включающее введение, номера и наименование всех разделов, подразделов, пунктов (если они имеют наименование), выводы и рекомендации, библиографический список, наименования всех приложений с указанием номеров страниц, с которых они начинаются. Слово «СОДЕРЖАНИЕ» записывают в виде заголовка (симметрично тексту).

В разделе «ВВЕДЕНИЕ» следует обосновать актуальность темы, охарактеризовать разрабатываемую проблему, ее научное, техническое и практическое значение, изложить историю вопроса и дать оценку современного состояния теории и практики. Актуальность темы раскрывается проблемами, решению которых способствует разрабатываемая информационная система.

Пояснительная записка к курсовому проекту состоит из четырех разделов основной части, каждый из которых соответствует одному этапу унифицированного процесса разработки ИС.

В первом разделе “Предпроектное обследование предметной области и спецификация требований к информационной системе” на основе общей информации о предметной области, в частности о фирме, как объекте хозяйственной деятельности, осуществляется создание диаграммы

активности, позволяющей выполнить моделирование бизнес-процессов учреждения, и осуществляется создание диаграммы вариантов использования (use case diagrams), позволяющей специфицировать функциональные требования к создаваемой системе.

Сначала приводится краткое текстовое описание предметной области. В описание включается описание внешней среды и описание предприятия взаимодействующего с внешней средой. В описании отражаются основные аспекты предметной области: документы, функции, сеть, люди, операционное время и цели (в минимальном составе: бизнес-процессы и отношения между объектами предметной области). Причем, описывать необходимо только те проблемы, которые решаются с помощью проектируемой ИС. Допускается включение иллюстраций, схем и т.п., раскрывающих текстовое описание.

Затем текстовое описание необходимо представить в виде UML-диаграммы активности, моделирующей алгоритмы ключевых процессов предметной области.

Далее концепция проектируемой ИС представляется текстовым описанием модели требований к ИС (прообраз технического задания на разработку ИС). Функциональные требования являются ключевым компонентом модели требований. Если следовать положениям RUP, то модель требований есть результат процесса разработки требований, в ходе которого разрабатывается бизнес-модель (модель предметной области), определяются бизнес - процессы и только затем формируются требования к системе в виде вариантов использования(прецедентов) ИС.

Рекомендации по созданию диаграмм вариантов использования:

- не пытаться записать все требования в виде вариантов использования или разложить всю систему на варианты использования;
- писать варианты использования кратко и ёмко, избегать тривиальных вариантов использования;

-лучше ориентироваться на обычные списки или таблицы (действующие лица, их цели и названия вариантов использования);

-варианты использования желательно сгруппировать по главному действующему лицу, а далее – по его цели, если их несколько;

-не смешивать варианты использования и пользовательский интерфейс.

Вариант использования – это не текстовая документация интерфейса. (Хотя и такая трактовка Варианта использования имеет право на существование). Пользовательский интерфейс часто предлагает список объектов (к примеру, товаров), который пользователь может просматривать (фильтровать, искать), выбирать отдельный объект, смотреть его, редактировать, удалять или же добавлять новые объекты. Всё это стандартные понятные функции. Нет нужны выписывать Вариант использования «Добавить товар», «Изменить товар», «Удалить товар» и т.д.;

-писать следует всё в активном залоге («Система отправляет письмо», «Пользователь выбирает товар», но не «Письмо отправляется», «Товар выбирается»). Если формулировка не получается, это верный знак, что вы что-то забыли смоделировать.

Рассмотрим пример формирования варианта использования. Допустим, мы имеем дело с системой закупок. Наш пользователь – Покупатель – сотрудник отдела снабжения. У него есть Начальник, который должен одобрить заказы своих подчиненных, прежде чем те будут выполнены. Основными действиями при этом являются:

-покупатель выбирает нужный товар и заказывает его;

-система помещает заказ в очередь и уведомляет Начальника отдела снабжения;

-начальник, получив уведомление от системы, проверяет заказ и одобряет его;

-система принимает заказ и высылает подтверждение Покупателю.

Эти четыре действия составляют не четыре, а один вариант использования: Заказ товара.

Пример диаграммы вариантов использования “Заказ товара”, выполненной в Rational Rose, представлен на рисунке 5.

В заключении этого раздела следует кратко указать основные нефункциональные требования, которые представляются в виде текстового описания ограничений среды или реализации, производительности, зависимости от платформы, надежности, расширяемости, режимов работы и т.п.

Во втором разделе “Эскизное проектирование информационной системы” осуществляется построение модели анализа. Она включает основные классы, необходимые для реализации выделенных вариантов использования, а также возможные связи между классами. Выделяемые классы разбиваются на три разновидности — интерфейсные, управляющие и классы данных. Эти классы представляют собой набор сущностей, в терминах которых работа системы должна представляться пользователям. Они являются понятиями, с помощью которых достаточно удобно объяснять себе и другим происходящее внутри системы, не слишком вдаваясь в детали.

Интерфейсные классы (boundary classes) соответствуют устройствам или способам обмена данными между системой и ее окружением, в том числе пользователями. Классы данных (entity classes) соответствуют наборам данных, описывающих некоторые однотипные сущности внутри системы. Эти сущности являются абстракциями представлений пользователей о данных, с которыми работает система. Управляющие классы (control classes) соответствуют алгоритмам, реализующим какие-то значимые преобразования данных в системе и управляющим обменом данными с ее окружением в рамках вариантов использования.

В курсовом проекте осуществляется построение диаграмм коммуникации и последовательности.



Прежде всего, на диаграмме коммуникации(кооперации) в виде прямоугольников изображаются участвующие во взаимодействии объекты, содержащие имя объекта, его класс и, возможно, значения атрибутов. Далее, указываются ассоциации между объектами в виде различных соединительных линий. При этом можно явно указать имена ассоциации и ролей, которые играют объекты в данной ассоциации. Дополнительно могут быть изображены динамические связи - потоки сообщений. Они представляются также в виде соединительных линий между объектами, над которыми располагается стрелка с указанием направления, имени сообщения и порядкового номера в общей последовательности инициализации сообщений.

Диаграмма последовательности должна моделировать перечень ключевых функций системы (т.н. ответственностей), которые суть действия, замещающие в предметной области ручные действия, то есть автоматизирующие труд пользователей.

UML диаграмма последовательности по существу отражает модель поведения(модель взаимодействия объектов) системы. Модель поведения разрабатывается для каждой ключевой функции ИС (варианта использования).

В третьем разделе “Техническое проектирование информационной системы” создается модель проектирования. Модель проектирования является детализацией и специализацией модели анализа. Она также состоит из классов, но более четко определенных, с более точным и детальным распределением обязанностей, чем классы модели анализа. Классы модели проектирования должны быть специализированы для конкретной используемой платформы.

Модель проектирования информационной системы в курсовом проекте включает в себя следующие диаграммы:

-классов (class diagrams) – для моделирования статической структуры классов подсистемы и связей между ними;

-состояний (State machine) – для моделирования поведения объектов подсистемы при переходе из одного состояния в другое.

Диаграмма состояний в работе должна описывать все возможные состояния одного экземпляра определенного класса и возможные последовательности его переходов из одного состояния в другое, то есть моделирует все изменения состояний объекта как его реакцию на внешние воздействия.

В четвертом разделе “Рабочее проектирование информационной системы”, осуществляется разработка модели реализации информационной системы.

Разработка модели реализации информационной системы в курсовом проекте включает построение следующих диаграмм:

- компонентов (component diagrams) – для моделирования иерархии компонентов подсистемы;

-развертывания(размещения) - для определения аппаратной архитектуры системы.

Под моделью реализации в рамках RUP и UML понимают набор компонентов результирующей системы и связей между ними. Под компонентом здесь имеется в виду компонент сборки — минимальный по размерам кусок кода системы, который может участвовать или не участвовать в определенной ее конфигурации, единица сборки и конфигурационного управления. Связи между компонентами представляют собой зависимости между ними. Если компонент зависит от другого компонента, он не может быть поставлен отдельно от него. Часто компоненты представляют собой отдельные файлы с исходным кодом.

Диаграмма развертывания представляет собой набор узлов системы, являющихся физически отдельными устройствами, которые способны

обрабатывать информацию — серверами, рабочими станциями, принтерами, контроллерами датчиков и пр., со связями между ними, образованными различного рода сетевыми соединениями. Каждый узел может быть нагружен некоторым множеством компонентов, определенных в модели реализации.

Цель построения модели развертывания — определить физическое положение компонентов распределенной системы, обеспечивающее выполнение ею нужных функций в тех местах, где эти функции будут доступны и удобны для пользователей.

В разделе «ЗАКЛЮЧЕНИЕ» должен содержаться анализ результатов выполненной работы, оценка полноты решения поставленных задач, рекомендации по конкретному использованию результатов работы, ее экономическая, научная, социальная значимость.

Список использованных источников включает в себя пособия, справочники, каталоги, прейскуранты, стандарты, инструкции, альбомы, статьи, патенты, иностранные источники и т.п. Список составляется в порядке появления ссылок на источники в тексте пояснительной записки.

Приложения включают материалы иллюстрационного и вспомогательного характера: таблицы, рисунки большого формата, дополнительные расчеты, листинги кода и т.п. Приложения оформляют как продолжение пояснительной записки на последующих ее листах или в виде самостоятельного документа.

## **2.3 Оформление пояснительной записки**

Оформление пояснительной записки выполняют в соответствии с требованиями Правил оформления и требований к содержанию курсовых проектов (работ) и выпускных квалификационных работ [4].

Пояснительную записку выполняют одним из следующих способов:

- рукописным четким, разборчивым почерком с высотой букв и цифр не менее 2,5 мм. Текст должен быть написан тушью, чернилами или пастой синего или черного цвета. Расстояние между строк 8–10 мм;

-любым печатным способом на пишущей машинке или с использованием компьютера и принтера на одной стороне листа белой бумаги формата А4 (210x297 мм) по ГОСТ 9327 через полтора интервала, шрифт Times New Roman, размер букв шрифта 14, цвет черный.

При оформлении курсового проекта соблюдать следующие размеры полей: правое – 10 мм, левое – 30 мм, верхнее – 20 мм, нижнее – 20 мм.

Номер листа проставляется в центре нижней части листа без точки. На титульном листе и на листе задания номер не проставляется.

Допускается оформление пояснительной записки с рамками и основными надписями по форме 2 и 2а ГОСТ 2.104 и формам 9 и 9а ГОСТ 2.106. При этом номер листа проставляется в соответствующей графе основной надписи.

Основную надпись по форме 2 необходимо выполнять только на листе «СОДЕРЖАНИЕ», а последующие листы выполнять с основной надписью по форме 2а.

Нумерация страниц пояснительной записки и приложений, входящих в состав записки, должна быть *сквозная*.

Основная часть пояснительной записки разделяется на разделы и подразделы.

Разделы должны иметь порядковые номера в пределах всего документа, обозначенные арабскими цифрами без точки. Подразделы должны иметь нумерацию в пределах каждого раздела. Номер подраздела состоит из номеров раздела и подраздела, разделенных точкой. В конце номера подраздела точка не ставится.

Подразделы могут состоять из нескольких пунктов. Нумерация пунктов должна быть в пределах подраздела, и номер пункта должен состоять из номеров раздела, подраздела и пункта, разделенных точками.

Каждый раздел, подраздел, пункт, подпункт и перечисление записывают с абзацного отступа. Абзацный отступ должен быть одинаковым по всему тексту пояснительной записки и равен 12–12.5 мм).

Все разделы и подразделы должны иметь заголовки. Пункты, как правило, заголовков не имеют.

Заголовки должны четко и кратко отражать содержание разделов, подразделов.

Заголовок раздела(подраздела) печатают, отделяя от номера пробелом, начиная с прописной буквы, не приводя точку в конце и не подчеркивая. При этом номер раздела (подраздела) печатают после абзацного отступа.

Заголовки разделов выделяют полужирным шрифтом.

Переносы частей слов в заголовках не допускаются. Если заголовок состоит из двух и более предложений, их разделяют точкой.

В заголовках следует избегать сокращений (за исключением общепризнанных аббревиатур, единиц величин и сокращений, входящих в условные обозначения продукции).

Расстояние между заголовком раздела и текстом, между заголовком раздела и заголовком подраздела должно быть равно 12÷15 мм, между заголовком подраздела и текстом должно быть 7÷8 мм.

При переносе текста на следующую страницу после наименования раздела (подраздела) рекомендуется записать не менее двух строк.

В курсовом проекте допускается последующий раздел оформлять не с нового листа, а в продолжение текста предыдущего раздела.

Текст пояснительной записки должен быть кратким, четким и не допускать различных толкований. Изложение текста должно быть от третьего лица. При изложении обязательных требований в тексте должны

применяться слова «должен», «следует», «необходимо», «требуется», «чтобы», «разрешается только», «не допускается», «запрещается», «не следует». При изложении других положений следует применять слова – «могут быть», «как правило», «при необходимости», «может быть», «в случае» и т.д.

При этом допускается использовать повествовательную форму изложения текста документа, например, «применяют», «указывают» и т.п.

В пояснительной записке должны применяться научно-технические термины, обозначения и определения, установленные соответствующими стандартами, а при их отсутствии – общепринятые в научно-технической литературе.

Если в пояснительной записке принята специфическая терминология, то в конце его (перед библиографическим списком) должен быть перечень принятых терминов с соответствующими разъяснениями. Перечень включают в содержание пояснительной записки.

В тексте пояснительной записки не допускается:

- применять обороты разговорной речи;
- применять для одного и того же понятия различные научно-технические термины, близкие по смыслу (синонимы), а также иностранные слова и термины при наличии равнозначных слов и терминов в русском языке;
- применять произвольные словообразования;
- применять сокращения слов, кроме установленных правилами русской орфографии и соответствующих ГОСТ 7.12;
- сокращать обозначения единиц величин, если они употребляются без цифр, за исключением единиц величин в таблицах и в расшифровках буквенных обозначений, входящих в формулы и рисунки.

Опечатки, описки, графические неточности, обнаруженные в процессе выполнения работы, допускается исправлять аккуратным заклеиванием или

закрашиванием белой краской и нанесением на том же месте и тем же способом исправленного текста. Повреждение листов, помарки и следы не полностью удаленного текста не допускаются.

Все иллюстрации(диаграммы, схемы, компьютерные распечатки, фотографии), включаемые в текстовый материал пояснительной записки, именуется рисунками. При выполнении иллюстраций с помощью прикладных программ оформление их должно соответствовать требованиям стандартов.

Иллюстрации, за исключением иллюстраций приложений, следует нумеровать арабскими цифрами сквозной нумерацией. Если рисунок один, то он обозначается «Рисунок 1». Слово «Рисунок» и его наименование располагают посередине строки.

Допускается нумеровать иллюстрации в пределах раздела. В этом случае номер иллюстрации состоит из номера раздела и порядкового номера иллюстрации, разделенных точкой. Например, Рисунок 1.1.

При ссылках на иллюстрации следует писать:

- при сквозной нумерации – «в соответствии с рисунком 2»;
- при нумерации в пределах раздела – «в соответствии с рисунком 1.2»;
- при ссылках на иллюстрации, помещенные в приложении – «в соответствии с рисунком Б.4».

Иллюстрации, при необходимости, могут иметь наименование и пояснительные данные (подрисуночный текст). Слово «Рисунок» и наименование помещают после пояснительных данных симметрично рисунку. Примеры представлены в разделе 1 данных методических рекомендаций.

Оформление списка использованных источников осуществляется в соответствии с общими требованиями и правилами составления библиографического описания документа, представленными в ГОСТ Р 7.0.5–2008 «Библиографическая ссылка. Общие требования и правила

составления», введенного в действие с 1 января 2009 г. По ГОСТ Р 7.0.5–2008 допускается не использовать тире между областями библиографического описания. Список использованных источников начинают с нового листа. Слова «СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ» оформляются симметрично основному тексту.

Примеры оформления списка находятся в ГОСТ 7.0.5 и в списке использованных источников данных методических рекомендаций.

Приложения оформляют как продолжение пояснительной записки на последующих ее листах или в виде самостоятельного документа. Слово «ПРИЛОЖЕНИЕ» печатается в верхней части страницы, посередине, все буквы прописные, выделяются полужирным шрифтом, под ним в скобках указывается статус приложения, используются слова «обязательное», «рекомендуемое» или «справочное».

## **2.4 Рекомендуемая литература для использования**

В процессе курсового проектирования необходимо использовать различные виды научно-технической литературы по теме курсового проекта. При этом важно, чтобы используемые источники информации отражали современное состояние средств и методов разработки информационных систем. Общая рекомендация заключается в использовании литературы с годом издания не более пятилетней давности. Среди наиболее популярных изданий можно выделить следующие.

1. Исаев Г.Н. Проектирование информационных систем: учебное пособие/Г.Н.Исаев, М.:Издательство "Омега-Л", 2013.424с.

2. Денищенко Г. Проектирование информационных систем изд.2, испр. /Г. Денищенко, В.Грекул, Н. Коровкина. М.: БИНОМ, ТОРГОВЫЙ ДОМ, БИНОМ. Лаборатория знаний, Бином-Пресс, ИНТУИТ.РУ, 2008. 736с.



3. Балдин К.В. Информационные системы в экономике : учебник / К.В. Балдин, В.Б. Уткин. - 3-е изд. М. : Дашков и Ко, 2012. 394
4. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования./Э.Гамма, Р. Хелм, Р. Джонсон, Дж. Влссидес. СПб.: Питер, 2007. 366 с.
5. Леоненков, А. В. UML 2: Объектно-ориентированный анализ и проектирование; Моделирование программных систем; Нотация и семантика последней версии UML 2 и др. / А. В. Леоненков. СПб.: БХВ-ПЕТЕРБУРГ, 2007. 576 с.
6. Липаев, В. В. Системное проектирование сложных программных средств для информационных систем. /В. В. Липаев. СПб.: Синтег, 2002 . 268с.
7. Лясников И. Кризис открыл дорогу малому и среднему бизнесу на рынке профессионального софта. URL: [http://www. pr@lexpro.ru](http://www.pr@lexpro.ru). – М.: LEXPRO Soft. - 2008.-4 с.
8. Бобровский, С.И. Delphi 2006. Новые возможности./С.И. Бобровский. СПб.: Питер, 2006. 288 с.
9. Архитектура информационных систем. Максим Смирнов URL: <http://mxsmirnov.wordpress.com/>
10. Распределенные объектные технологии в информационных системах . URL: <http://www.unislabs.com/>
11. Об архитектуре программных и информационных систем. URL:<http://www.fostas.ru/about/arch.php>
12. Данилин А., Архитектура и стратегия. «Инь» и «янь» информационных технологий. Интернет-университет информационных технологий. /А.Данилин, А.Слюсаренко. ИНТУИТ.ру, 2005 г. 238с.Учебник свободно доступен на сайте URL: <http://www.intuit.ru>.
13. Орлов, С. Технология разработки программного обеспечения: учебник. СПб.: Питер, 2012. 348 с.

14. Гурьянов В. И. Имитационное моделирование на UML SP: монография / В.И. Гурьянов. Чебоксары : Филиал СПбГЭУ в г. Чебоксары, 2014. 135 с. URL: <http://www.guryanov-sym-uml-sp.ru/>

15. Колесов Ю.Б., Сениченков Ю.Б. Моделирование систем. Объектно-ориентированный подход. СПб: БХВ-Петербург, 2009. - 268с.

Представленные источники информации имеют фундаментальный характер. На практике иногда целесообразно использовать научные статьи, аналитические обзоры и другие публикации журналов: Программные продукты и системы, Информационные технологии, Компьютер пресс и другие.

### **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. ГОСТ Р ИСО/МЭК 15288—2005. «Информационная технология. Системная инженерия. Процессы жизненного цикла систем». М.: 2006.

2. Исаев, Г.Н. Проектирование информационных систем: учебное пособие. /Г.Н.Исаев,-М.:Издательство "Омега-Л", 2013.- 424с.

3. Трофимов, С. CASE-технологии: Практическая работа в Rational Rose. Изд. 2-е. /С. Трофимов. М.: Бинوم-Пресс, 2002. - 288с.

4. Правила оформления и требования к содержанию курсовых проектов (работ) и выпускных квалификационных работ. Редакция 4. Введ. 30.12.2015. - Ростов-на-Дону: ДГТУ, 2015. - 83с.



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Технологический институт сервиса (филиал) ДГТУ в г.Ставрополе  
(ТИС (филиал) ДГТУ в г.Ставрополе)**

## **МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

по выполнению курсовых работ (проектов)  
по дисциплине «Модели и методы проектирования  
информационных систем»  
для студентов направления подготовки  
09.04.02 Информационные системы и технологии  
Направленность (профиль) Информационные системы и  
технологии

## Содержание

Введение

Цель, задачи и реализуемые компетенции

Формулировка задания

Структура работы

Общие требования к написанию и оформлению работы

Последовательность выполнения задания

Критерии оценивания работы

Порядок защиты работы

Список рекомендуемых информационных источников

## **Введение**

В настоящих методических указаниях представлена методика написания курсовой работы по дисциплине «Модели и методы проектирования информационных систем» для студентов, направления 09.03.02 «Информационные системы и технологии». Методические указания по проведению к курсовой работе разработаны в соответствии с рабочей программой учебной дисциплины.

Курсовая работа позволяет развить творческий потенциал студентов и подготовить их к выполнению выпускной квалификационной работы. Главное назначение курсовой работы по дисциплине «Модели и методы проектирования информационных систем» состоит в подготовке студентов к самостоятельному выполнению исследовательской работы, связанной с проблемами в области экономики.

### **Цель, задачи и реализуемые компетенции**

Написание курсовой работы - составная часть учебного процесса, предполагающая самостоятельную работу студента. Цели и задачи курсовой работы разнообразны: научная, познавательная, учебная, методическая. Данные цели проявляются через следующие конкретные задачи курсовой работы:

- систематизация, закрепление и расширение теоретических знаний, полученных на лекциях и семинарах по экономической безопасности;
- привитие навыков самостоятельной работы по подбору литературы, работы с научной литературой;
- умение самостоятельно систематизировать и изложить знания, полученные в процессе самостоятельного изучения литературы;
- привитие навыков научно-исследовательской работы, использование анализа и самостоятельных выводов по экономическим проблемам;
- углубленное изучение наиболее актуальных экономических проблем, уяснение связи теории с практикой в обеспечении экономической безопасности.

В результате освоения данной дисциплины формируются следующие компетенции у обучающегося:

УК-2.2: Анализирует альтернативные варианты реализации проекта, определяет целевые этапы, основные направления работы.

ПК-4.2: Анализирует информацию для инициации и планирования проекта в области информационных технологий.

Курсовая работа должна свидетельствовать о способности автора к систематизации и расширению полученных теоретических знаний, о

владении практическими навыками по дисциплине «Модели и методы проектирования информационных систем», о сформированности компетенций при решении поставленных в работе задач.

Курсовая работа должна привить студенту навыки творческого изучения и решения профессиональных задач.

Проведенное исследование может касаться чисто теоретической проблемы или ориентироваться на решение практических задач, связанных с дальнейшей профессиональной деятельностью студента.

### **Формулировка задания**

Во **введении** обосновывается актуальность темы и ее практическая значимость, выделяются цели и задачи курсовой работы.

Здесь же оговаривается объект исследования, круг исследуемых вопросов, обозначается фактический материал, на котором строится исследование. Введение содержит описание применяемых методов исследования и т.п.

**Первая глава**, как правило, носит теоретический характер. В ней освещается история исследуемого вопроса, существующий в литературе дискуссионный материал, должна быть рассмотрена законодательная и нормативная база.

При осмыслении литературной дискуссии студент должен не только отразить имеющиеся в литературе точки зрения, но и обозначить и аргументировать свою позицию по рассматриваемому вопросу. Представить по возможности имеющийся зарубежный опыт решения проблемы.

Во **второй главе** студент анализирует статистический материал, который служит базой для выводов и предложений студента.

Недопустимо простое пересказывание существующих методик, специальной литературы, учебников. Студент должен творчески подойти к рассматриваемому вопросу и обосновать свою точку зрения.

В **третьей главе** на базе исследования, проведенного в предыдущих главах, приводятся конкретные решения задачи, предлагаемые студентом. Основная часть предложений должна быть связана с обеспечением экономической безопасности организации, либо ее составляющих (например, финансовой безопасности организации), либо с нейтрализацией возможных угроз экономической безопасности.

Предлагаемые решения должны сопровождаться иллюстративным материалом (графики, диаграммы, расчеты и т.п.).

Особое внимание следует уделить последствиям внедрения предложений студента на практику соответствующей деятельности, на изменение соответствующих экономических показателей деятельности предприятия в случае осуществления им мероприятий, предложенных студентом.

Все выводы и рекомендации, предлагаемые в работе, должны быть обоснованы и аргументированы.

В заключении кратко упоминаются основные этапы исследования, отражаются основные результаты, полученные студентом, требуется сформулировать основные выводы и рекомендации об эффективности производства и мероприятий, направленных на обеспечение экономической безопасности организации.

### **Структура работы**

Независимо от избранной темы рекомендуется придерживаться приведенной ниже структуры курсовой работы:

Титульный лист.

Содержание.

Задание.

Введение.

Основной текст работы.

Расчетная часть

Заключение.

Список использованных источников.

Приложения.

Основной текст работы должен состоять из введения, трех глав и заключения. Каждая глава разбивается на два-три параграфа.

Дальнейшее дробление глав и параграфов не допускается.

Объем дипломной работы студента должен составлять 35-50

страниц текста, набранного на компьютере (без учета приложений).

Примерный объем структурных частей дипломной работы (в процентах к общему объему основного текста):

- введение -5%,
- основная часть - 50%,
- расчетная часть - 40%,
- заключение - 5%.

### **Общие требования к написанию и оформлению работы**

Текст работы должен быть машинописным на одной стороне белой бумаги формата А4 (210x290 мм). Поля должны оставаться по всем четырем сторонам листа. Размер левого поля - 30 мм, правого - 10 мм, верхнего и нижнего полей - 20 мм.

Шрифт принтера должен быть четким, черного цвета. При выполнении работы на компьютере (в текстовом редакторе WORD) необходимо установить следующие параметры:

- шрифт Times New Roman, размер шрифта 14;  
- выравнивание по ширине, первая строка равна отступ 1,25 см, межстрочный интервал – 1,5 (при этом отступ справа/слева и интервал перед/после равны нулю).

Работа должна быть сброшюрована.

Фамилии, названия учреждений, организаций, фирм, название изделий и другие имена собственные в работе приводятся на языке оригинала.

Все страницы работы, включая иллюстрации и приложения, нумеруются по порядку без пропусков и повторений арабскими цифрами. Первой страницей считается титульный лист, на котором нумерация страниц не ставится. Номер страницы представляют в центре нижней части листа без точки.

Главы должны иметь порядковые номера в пределах всей работы, обозначенные арабскими цифрами без точки.

Каждый структурный элемент, в том числе главу работы необходимо начинать с нового листа (страницы). Название каждой главы записывают по центру прописными буквами с новой страницы без переносов, без точки в конце, не подчеркивая.

Название пункта, подпункта записывают с абзацного отступа 1,25, выравнивая по ширине. Заголовки подпунктов следует печатать с прописной буквы без точки в конце, не подчеркивая.

Расстояние между заголовками главы и пункта, подпункта - 1 свободная строка. Расстояние между заголовками и текстом – 1 свободная строка, между текстом и заголовком следующего пункта, подпункта – 2 свободные строки.

Формулы, за исключением формул, помещаемых в приложении, должны нумероваться сквозной нумерацией арабскими цифрами, которые записывают на уровне формулы справа в круглых скобках.

Ссылки в тексте на порядковые номера формул дают в скобках, например, ... в формулах (5, 6).

$$p = \frac{(\text{ЧП}_0 - \text{ЧП}) * \gamma(z) + \sqrt{D\text{ЧП}} * \beta(z)}{\text{ЧП}} * 100, \quad (5)$$

$$z = \frac{(\text{ЧП}_0 - \text{ЧП})}{\sqrt{D\text{ЧП}}}, \quad (6)$$

где  $p$  - риск инвестиционного проекта;

$\text{ЧП}_0$  - планируемая величина чистой прибыли, руб.;

$\text{ЧП}$  - среднее значение чистой прибыли при различных вариантах, руб.;

$D\text{ЧП}$  - дисперсия чистой прибыли;

$\gamma(z)$  - функция Лапласа;

$\beta(z)$  - функция Гаусса.

Таблицы применяют для лучшей наглядности и удобства сравнения показателей. Название таблицы, при его наличии, должно отражать ее





предоставляется право предложения своей тематики с необходимым обоснованием целесообразности ее разработки (Приложение А).

Вариант расчетной части определяется по первой букве фамилии студента (Таблица 1).

Таблица 1 - Варианты расчетной части курсовой работы

ВАРИАНТ	Начальная буква фамилии студента
1	А, Ю, Я, Е, Ж,
2	Б, Э, М, З, И, Ф,
3	В, Р, Ц, Д, Ш, Ч,
4	Г, П, Щ, Л, С,
5	Н, Т, О, К, У, Х.

Преподаватель призван оказывать научно-методическую помощь студенту при самостоятельном решении им научных и практических вопросов в творческой работе, связанной с поиском новых идей. Однако за результаты проделанной работы отвечает автор-студент.

Процесс выполнения курсовой работы имеет следующие этапы

- выбор темы;
- подбор и изучение литературы по теме исследования;
- сбор информации;
- обработка информации;
- составление плана курсовой работы;
- разработка программы исследования по разделам;
- составление инструментария (макетов таблиц);
- написание чернового варианта работы;
- редактирование и оформление работы в соответствии с требованиями;
- сдача курсовой работы на проверку;
- изучение замечаний и их доработка;
- защита курсовой работы.

Для успешного выполнения и защиты курсовой работы следует учесть следующие положения:

- курсовая работа выполняется в течение изучения курса дисциплины;
- преподавателю на проверку курсовой работы требуется десять дней;
- работа над замечаниями требует от 2 до 10 дней.

После проверки работы преподаватель сообщает дату ее защиты.

Курсовая работа, считается допущенной к защите, если она представлена в завершённом виде, подписана автором и имеет отметку преподавателя о ее допуске к защите.

### **Критерии оценивания работы**

Оценка по защите курсовой работы определяется оценками: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

**Оценка «отлично»** выставляется при условии, что:

- работа выполнена самостоятельно, носит творческий характер, возможно содержание элементов научной новизны;
- собран, обобщен и проанализирован достаточный объем литературных источников;
- при написании и защите работы студентом дневного отделения продемонстрирован высокий уровень развития общекультурных и профессиональных компетенций, теоретические знания и наличие практических навыков;
- работа хорошо оформлена и своевременно представлена на кафедру, полностью соответствует требованиям, предъявляемым к содержанию и оформлению курсовых работ;
- на защите освещены все вопросы исследования, ответы студента на вопросы профессионально грамотны, исчерпывающие, результаты исследования подкреплены статистическими критериями;

**Оценка «хорошо»** ставится, если:

- тема работы раскрыта, однако выводы и рекомендации не всегда оригинальны и / или не имеют практической значимости, есть неточности при освещении отдельных вопросов темы;
- собран, обобщен и проанализирован необходимый объем психологической литературы, но не по всем аспектам исследуемой темы сделаны выводы и обоснованы практические рекомендации;
- при написании и защите работы студентом продемонстрирован средний уровень развития общекультурных и профессиональных компетенций, наличие теоретических знаний и достаточных практических навыков;
- работа своевременно представлена на кафедру, есть отдельные недостатки в ее оформлении;
- в процессе защиты работы были неполные ответы на вопросы.

**Оценка «удовлетворительно»** ставится, когда:

- тема работы раскрыта частично, но в основном правильно, допущено поверхностное изложение отдельных вопросов темы;
- в работе недостаточно полно была использована психологическая литература, выводы и практические рекомендации не отражали в достаточной степени содержание работы;
- при написании и защите работы студентом продемонстрирован удовлетворительный уровень развития общекультурных и профессиональных компетенций, поверхностный уровень теоретических знаний и практических навыков;
- работа своевременно представлена на кафедру, однако не в полном объеме по содержанию и / или оформлению соответствует предъявляемым требованиям;

- в процессе защиты выпускник недостаточно полно изложил основные положения работы, испытывал затруднения при ответах на вопросы.

**Оценка «неудовлетворительно»** ставится, если:

- содержание работы не раскрывает тему, вопросы изложены бессистемно и поверхностно, нет анализа практического материала, основные положения и рекомендации не имеют обоснования;

- работа не оригинальна, основана на компиляции публикаций по теме;

- при написании и защите работы студентом продемонстрирован неудовлетворительный уровень развития общекультурных и профессиональных компетенций;

- работа несвоевременно представлена на кафедру, не в полном объеме по содержанию и оформлению соответствует предъявляемым требованиям;

- на защите студент дневного отделения показал поверхностные знания по исследуемой теме, отсутствие представлений об актуальных проблемах по теме работы, плохо отвечал на вопросы.

Руководитель имеет право рекомендовать лучшие работы для использования в учебном процессе и для внедрения их в практику. В тех случаях, когда работа признается неудовлетворительной, заведующий кафедрой устанавливает, может ли студент представить ее к повторной защите с доработкой или же обязан разработать новую тему, которую рекомендует кафедра.

В случае заимствования работы у других студентов и в случае несоответствия работы предъявляемым требованиям преподаватель оставляет за собой право не допускать курсовую работу к защите. В этом случае магистр обязан в течение недели представить новую курсовую работу по другому варианту, указанному преподавателем. В противном случае студент будет считаться не выполнившим курсовую работу.

Требуемый уровень оригинальности - не менее 50%.

### **Порядок защиты работы**

Получив от научного руководителя допуск к защите, студент начинает готовиться к защите курсовой работы, т.е. демонстрации знаний темы, умения отстаивать изложенный материал, аргументировать свои выводы и предложения.

По усмотрению научного руководителя процедура защиты курсовой работы может носить характер двустороннего взаимодействия (преподаватель – студент), а может быть и публичной, проходить в студенческой группе.

На защите студент должен кратко изложить содержание своей работы, поставленные в ней проблемы, привести сведения об источниках, на основе которых она написана. Желательно сделать презентацию работы в редакторе Power Point или PDF.

Процедура защиты предполагает устную форму ответов студента на вопросы, задаваемые научным руководителем по теме курсовой работы, поэтому студент должен подготовиться к вопросам, которые могут быть заданы по теме исследования.

Если студент хорошо подготовился к защите и дал исчерпывающие ответы на вопросы, учел замечания, содержащиеся в отзыве, а также ответил и на дополнительные вопросы научного руководителя, окончательная оценка курсовой работы может быть повышена по сравнению с первоначальной (предварительной) оценкой, отраженной в отзыве. И наоборот, если в процессе защиты студент показывает слабое знание рассматриваемых в курсовой работе вопросов или не ориентируется в собственной курсовой работе, то оценка может быть снижена вплоть до неудовлетворительной.

К текущей сессии студент допускается только после получения положительной оценки за курсовую работу по данной учебной дисциплине.

### СПИСОК РЕКОМЕНДУЕМЫХ ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

<b>6.1.1. Основная литература</b>				
	Авторы,	Заглавие	Издательство, год	Адрес
Л1.1	Бурков А. В.	Проектирование информационных систем в Microsoft SQL Server 2008 и Visual Studio 2008	Москва: Интернет - Университет Информационных Технологий (ИНТУИТ), 2016	<a href="http://www.iprbookshop.ru/52166.html">http://www.iprbookshop.ru/52166.html</a>
Л1.2	Краюткина, Е. В.	Методы и средства проектирования информационных систем и технологий: учебное пособие	Ставрополь: Северо-Кавказский федеральный университет, 2015	<a href="http://www.iprbookshop.ru/62959.html">http://www.iprbookshop.ru/62959.html</a>
<b>6.1.2. Дополнительная литература</b>				
	Авторы,	Заглавие	Издательство, год	Адрес
Л2.1	Фельдман Я. А.	Создаем информационные системы	Москва: СОЛЮН-ПРЕСС, 2009	<a href="http://www.iprbookshop.ru/8684.html">http://www.iprbookshop.ru/8684.html</a>
Л2.2	Жданов, С. А., Соболева, М. Л., Алфимова, А. С., Матросов, В. Л.	Информационные системы: учебник для студентов учреждений высшего образования	Москва: Прометей, 2015	<a href="http://www.iprbookshop.ru/58132.html">http://www.iprbookshop.ru/58132.html</a>
<b>6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"</b>				
Э1	Задания на лабораторные работы			
Э2	Задания на практические занятия			
Э3	Краюткина Е.В. Методы и средства проектирования информационных систем и технологий [Электронный ресурс]: учебное пособие/ Краюткина Е.В.— Электрон. текстовые данные.— Ставрополь: Северо-Кавказский федеральный университет, 2015.— 152 с.— Режим доступа: <a href="http://www.iprbookshop.ru/62959.html">http://www.iprbookshop.ru/62959.html</a> .— ЭБС «IPRbooks»			
Э4	Модели информационных систем [Электронный ресурс]: учебное пособие/ В.П. Бубнов [и др.].— Электрон. текстовые данные.— М.: Учебно-методический центр по образованию на железнодорожном транспорте, 2015.— 188 с.— Режим доступа: <a href="http://www.iprbookshop.ru/45279.html">http://www.iprbookshop.ru/45279.html</a> .— ЭБС «IPRbooks»			
Э5	Золотов С.Ю. Проектирование информационных систем [Электронный ресурс]: учебное пособие/ Золотов С.Ю.— Электрон. текстовые данные.— Томск: Томский государственный университет систем управления и радиоэлектроники, Эль Контент, 2013.— 88 с.— Режим доступа: <a href="http://www.iprbookshop.ru/13965.html">http://www.iprbookshop.ru/13965.html</a> .— ЭБС «IPRbooks»			

Рекомендуемая тематика курсовых работ

Разработка плана проекта информационной системы (для предметной области магистерской диссертации)

Разработка плана проекта программного обеспечения (для предметной области магистерской диссертации)

Разработка плана проекта информационной сети (для предметной области магистерской диссертации)